

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

«До захисту допущено»

Науковий керівник кафедри

\_\_\_\_\_ Іван ДИЧКА

«\_\_\_» \_\_\_\_\_ 2020 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інженерія програмного  
забезпечення комп'ютерних та інформаційно-пошукових систем»**

**спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Програмне забезпечення для аналізу рекламної кампанії»**

Виконав:

студент IV курсу, групи КП-61

Нечипоренко Владислав Олександрович \_\_\_\_\_

Керівник:

Старший викладач кафедри ПЗКС, к.т.н.,

Люшенко Леся Анатоліївна \_\_\_\_\_

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович \_\_\_\_\_

Рецензент:

Доцент кафедри ММСА ІПСА, к.ф.-м.н., доцент,

Шубенкова Ірина Анатоліївна \_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення комп'ютерних та інформаційно-пошукових систем»

«ЗАТВЕРДЖУЮ»

Науковий керівник кафедри

\_\_\_\_\_ Іван ДИЧКА

«\_\_» \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

Нечипоренку Владиславу Олександровичу

1. Тема проєкту «Програмне забезпечення для аналізу рекламної кампанії», керівник проєкту Люшенко Леся Анатоліївна, к.т.н., старший викладач, наказом по університету від «\_\_» \_\_\_\_\_ 2020 р. № \_\_\_\_\_
2. Термін подання студентом проєкту «\_\_» червня 2020 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
  - аналіз наявних рішень поставленого завдання;
  - обґрунтування вибору засобів реалізації;
  - розроблення мобільного застосунку;
  - аналіз мобільного застосунку.
5. Перелік обов'язкового графічного матеріалу:
  - алгоритм взаємодії графічних елементів (креслення);
  - схема бази даних (креслення);
  - діаграма варіантів використання (плакат);
  - дерево проблем системи (плакат).

## 6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

## 7. Дата видачі завдання «31» жовтня 2019 р.

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	14.11.2019	
2.	Розроблення та узгодження технічного завдання	28.11.2019	
3.	Розроблення структури мобільного застосунку	15.12.2019	
4.	Підготовка матеріалів першого розділу дипломного проєкту	30.12.2019	
5.	Розроблення дизайну сторінок та графічних елементів	03.02.2020	
6.	Підготовка матеріалів другого розділу дипломного проєкту	20.02.2020	
7.	Програмна реалізація мобільного застосунку	10.03.2020	
8.	Тестування мобільного застосунку	17.03.2020	
9.	Підготовка матеріалів третього розділу дипломного проєкту	30.03.2020	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	11.04.2020	
11.	Підготовка графічної частини дипломного проєкту	21.04.2020	
12.	Оформлення документації дипломного проєкту	26.05.2020	

Студент

Влад НЕЧИПОРЕНКО

Керівник проєкту

Леся ЛЮШЕНКО

## АНОТАЦІЯ

Цей дипломний проєкт присвячений розробленню програмного забезпечення для аналізу рекламної кампанії.

У роботі виконано обґрунтування та аналіз обраної тематики, аналіз наявних рішень поставленого завдання, обґрунтування вибору засобів реалізації, аналіз архітектури та структури програмного забезпечення та аналіз розробленого програмного забезпечення.

Програмне забезпечення реалізовано у вигляді мобільного застосунку та надає підприємствам можливість здійснювати систематизований збір та аналіз їхньої рекламної кампанії. Збір інформації відбувається з обраного джерела, після чого основна інформація відображається у вигляді графіків. Будь-який графік можна детально переглянути з додатковими налаштуваннями. Також підраховуються основні показники рекламної кампанії та аналізується популярність рекламної продукції в соціальних мережах та інтернеті. Після завершення рекламної кампанії всі показники аналізуються та виводиться загальна інформація.

Інформаційна безпека мобільного застосунку реалізована за рахунок надання прав доступу тільки зареєстрованим користувачам. Лише зареєстровані та авторизовані користувачі мають доступ до сторінок мобільного застосунку.

У цьому дипломному проєкті розроблено: архітектуру та структуру мобільного застосунку, алгоритм взаємодії графічних елементів, процедуру створення та перегляду аналізу рекламної кампанії, а також інтерфейс та дизайн мобільного додатку.

## **ABSTRACT**

This thesis project is devoted to the development of software for the analysis of the advertising campaign.

The paper substantiates and analyzes the selected topics, analyzes the available solutions to the problem, justifies the choice of means of implementation, analysis of the architecture and structure of the software and analysis of the developed software.

The software is implemented in the form of a mobile application and provides companies with the opportunity to systematically collect and analyze their advertising campaign. The collection of information is extracted from the selected source, after which the basic information is displayed in the form of graphs. Any schedule can be viewed in detail with additional settings. The main indicators of the advertising campaign are also calculated and the popularity of advertising products on social networks and the Internet is analyzed. At the end of the advertising campaign, all indicators are analyzed and general information is displayed.

Information security of the mobile application is implemented by granting access rights only to registered users. Only registered and authorized users have access to the pages of the mobile application.

In this diploma project the following is developed: architecture and structure of mobile application, algorithm of interaction of graphic elements, procedure of creation and viewing of the analysis of advertising campaign, and also the interface and design of the mobile application.

## **АННОТАЦИЯ**

Данный дипломный проект посвящен разработке программного обеспечения для анализа рекламной кампании.

В работе выполнено обоснование и анализ выбранной тематики, анализ имеющихся решений поставленной задачи, обоснование выбора средств реализации, анализ архитектуры и структуры программного обеспечения и анализ разработанного программного обеспечения.

Программное обеспечение реализовано в виде мобильного приложения и предоставляет предприятиям возможность осуществлять систематизированный сбор и анализ их рекламной кампании. Сбор информации происходит с выбранного источника, после чего основная информация отображается в виде графиков. Любой график можно детально просмотреть с дополнительными настройками. Также подсчитываются основные показатели рекламной кампании и анализируется популярность рекламной продукции в социальных сетях и интернете. После завершения рекламной кампании все показатели анализируются и выводится общая информация.

Информационная безопасность мобильного приложения реализована за счет предоставления прав доступа только зарегистрированным пользователям. Только зарегистрированные и авторизованные пользователи имеют доступ к страницам мобильного приложения.

В данном дипломном проекте разработаны: архитектура и структура мобильного приложения, алгоритм взаимодействия графических элементов, процедура создания и просмотра анализа рекламной кампании, а также интерфейс и дизайн мобильного приложения.

ДП.045440-01-90 Програмне забезпечення для аналізу рекламної кампанії.  
Відомість проекту

Позначення	Найменування	Кіл-ть	Примітка
	Документація проекту		
ДП.045440-02-91	Програмне забезпечення	5	
	для аналізу рекламної		
	кампанії.		
	Технічне завдання		
ДП.045440-03-81	Програмне забезпечення	70	
	для аналізу рекламної		
	кампанії.		
	Пояснювальна записка		
ДП.045440-04-51	Програмне забезпечення	4	
	для аналізу рекламної		
	кампанії.		
	Програма та методика		
	тестування		
ДП.045440-05-34	Програмне забезпечення	10	
	для аналізу рекламної		
	кампанії.		
	Керівництво користувача		
ДП.045440-06-99	Програмне забезпечення	1	
	для аналізу рекламної		
	кампанії.		
	Алгоритм взаємодії		
	графічних елементів.		
	Схема алгоритму		

[illegible]



**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

\_\_\_\_\_ Іван ДИЧКА

«\_\_\_» \_\_\_\_\_ 2019 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ДЛЯ АНАЛІЗУ РЕКЛАМНОЇ КАМПАНІЇ**

**Технічне завдання**

ДП.045440-02-91

«ПОГОДЖЕНО»

Керівник проєкту:

\_\_\_\_\_ Леся ЛЮШЕНКО

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Владислав НЕЧИПОРЕНКО

## ЗМІСТ

1. Найменування та галузь застосування.....	3
2. Підстава для розроблення.....	3
3. Призначення розробки.....	3
4. Вимоги до програмного продукту.....	3
5. Вимоги до проєктної документації.....	4
6. Етапи проєктування.....	5
7. Порядок тестування розробки.....	5

## **1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ**

**Назва розробки:** Програмне забезпечення для аналізу рекламної кампанії.

**Галузь застосування:** інформаційні технології.

## **2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ**

Підставою для розроблення є завдання на дипломне проектування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

## **3. ПРИЗНАЧЕННЯ РОЗРОБКИ**

Розробка призначена для використання підприємствами чи звичайними людьми в якості інформаційного забезпечення в сфері рекламних кампаній з метою надання користувачеві можливості аналізу рекламної кампанії.

## **4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ**

Програмне забезпечення повинно забезпечувати такі основні функції:

- 1) можливість аналізу рекламної кампанії;
- 2) можливість аналізу рекламної продукції;
- 3) можливість зберігати дані у базі даних;
- 4) можливість розширення структури програмного забезпечення;
- 5) можливість реєстрації та авторизації у програмному забезпеченні;
- 6) можливість детального перегляду графіків.

Розробку виконати на платформі iOS (мова програмування Swift) з використанням шаблону проектування MVC.

Додаткові вимоги:

- 1) наявність меню програмного забезпечення;
- 2) наявність анімованих кнопок;
- 3) наявність логотипу та назви програмного додатку;
- 4) наявність згоди на обробку персональних даних та користувацької згоди;
- 5) дизайн сторінок з використанням в якості базових білого та чорних кольорів.

## **5. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ**

У процесі виконання проекту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
  - «Алгоритм взаємодії графічних елементів»;
  - «Схема бази даних».

## **6. ЕТАПИ ПРОЄКТУВАННЯ**

Вивчення літератури за тематикою роботи.....	14.11.2019
Розроблення та узгодження технічного завдання.....	28.11.2019
Розроблення структури програмного забезпечення.....	15.12.2019
Розроблення дизайну сторінок та графічних елементів.....	03.02.2020
Програмна реалізація поставленого завдання.....	17.03.2020
Тестування програмного забезпечення.....	03.04.2020
Підготовка матеріалів текстової частини проєкту.....	28.04.2020
Підготовка матеріалів графічної частини проєкту.....	12.05.2020
Оформлення технічної документації проєкту.....	25.05.2020

## **7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ**

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування.

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

\_\_\_\_\_ Іван ДИЧКА

«\_\_» \_\_\_\_\_ 2020 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ДЛЯ АНАЛІЗУ РЕКЛАМНОЇ КАМПАНІЇ**

**Пояснювальна записка**

ДП.045440-03-81

«ПОГОДЖЕНО»

Керівник проєкту:

\_\_\_\_\_ Леся ЛЮШЕНКО

Нормоконтроль:

Виконавець:

\_\_\_\_\_ Микола ОНАЙ

\_\_\_\_\_ Владислав НЕЧИПОРЕНКО

2020

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	3
ВСТУП.....	6
1. АНАЛІЗ НАЯВНИХ РІШЕНЬ ПОСТАВЛЕНОГО ЗАВДАННЯ .....	7
1.1. Загальні положення та аналіз предметної галузі .....	7
1.2. Аналіз аналогічних застосунків.....	8
1.3. Актуальність розроблення .....	12
1.4. Вимоги до програмного забезпечення .....	13
1.5. Висновки до розділу.....	15
2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	16
2.1. Обґрунтування вибору ОС мобільного застосунку .....	16
2.2. Обґрунтування вибору мови програмування .....	18
2.3. Обґрунтування вибору мови програмування для серверу .....	21
2.4. Обґрунтування вибору системи керування базами даних.....	25
2.5. Шаблон проєктування MVC .....	27
2.6. Висновки до розділу.....	29
3. РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ .....	31
3.1. Загальний вигляд мобільного застосунку .....	31
3.2. Аналіз функціональних вимог до мобільного застосунку .....	32
3.3. Архітектура мобільного застосунку .....	33
3.4. Структура мобільного застосунку .....	37
3.5. Висновки до розділу.....	45
4. АНАЛІЗ МОБІЛЬНОГО ЗАСТОСУНКУ .....	45
4.1. Тестування мобільного застосунку .....	46
4.2. Порівняння мобільного застосунку з наявними аналогами .....	47
4.3. Пропозиції для майбутнього покращення мобільного застосунку ...	48
4.4. Висновки до розділу.....	49
ВИСНОВКИ .....	51
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	53
ДОДАТКИ.....	56

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

*Аналіз рекламної кампанії* – це систематизований збір і аналіз інформації для підвищення ефективності рекламних заходів і збільшення попиту на продукцію чи послугу в рамках обмеженого бюджету.

*ПЗ* – програмне забезпечення.

*БД* – база даних.

*UI* – інтерфейс користувача.

*ОС* – операційна система.

*Лід* – потенційний клієнт, тим чи іншим чином прореагувавший на маркетингову комунікацію.

*ROI* (від англ. *Return on Investment*), також відомий як *ROR* (від англ. *Rate of Return* – рентабельність) – фінансовий коефіцієнт, який ілюструє рівень прибутковості або збитковості бізнесу, з оглядом на суму зроблених в цей бізнес інвестицій.

*Pay per click* (з англ. – «платити за клік») – це рекламна модель, що працює в інтернеті, де рекламодавець розміщує рекламу на сайтах, і сплачує їхнім власникам за натискання користувачем на розміщений банер (текстовий або графічний) або «тіло» документа (класичний приклад – код формату clickunder).

*Вартість замовлення* (англ. *CPO – cost per order*), також звана *вартістю покупки* - показник продуктивності реклами.

*CPL – Cost-per-Lead*, являє собою модель оплати, де рекламодавець платить за лід.

*CPT, Cost per Thousand* або *CPM, Cost per mille*, *Ціна за тисячу контактів* – вартість 1000 контактів з аудиторією цього медіа.

*CPUI (cost per unique impression)* – це цінова модель, при якій оплата з рекламодавця передбачається через усі покази за винятком не унікальних.

*Рейтинг кліків* або *Клікабельність* (англ. *Click-through rate, CTR*) – відношення числа кліків на оголошення до числа його показів. Вимірюється



у відсотках. Використовується для виміру ефективності онлайн-реклами для певного вебсайту чи email-кампаній.

*Показник CPUC (cost per unique click)* – середня вартість унікального кліка.

*Показник CPUU (cost per unique user)* – вартість одного унікального відвідувача (користувача).

*CPV (cost per visit)* – вартість одного відвідування.

*CPO (cost per order)* – середня вартість замовлення.

*CPS (cost per sale)* – середні витрати на продаж.

*CPCr (cost per customer)* – середня вартість придбання одного покупця (замовника).

*Управління відносинами з клієнтами (англ. Customer relationship management (CRM), укр. сі-ар-ем)* – поняття, що охоплює концепції, котрі використовуються компаніями для управління взаємовідносинами зі споживачами, включно зі збором, зберіганням й аналізом інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

*Економічний ефект рекламування* – це різниця між прибутком, отриманого від додаткового товарообігу, викликаного рекламними заходами, і витратами на рекламу. Економічний ефект рекламних заходів може бути: позитивним-витрати на рекламу менше додаткового прибутку; негативним-витрати на рекламу вище додаткового прибутку; нейтральними-витрати на рекламу рівні додатковому прибутку.

*Модель–вигляд–контролер (або Модель–представлення–контролер, англ. Model-view-controller, MVC)* – архітектурний шаблон, який використовується під час проєктування та розроблення програмного забезпечення.

*Модульне тестування (англ. Unit testing)* – це метод тестування програмного забезпечення, який полягає в особному тестуванні кожного модуля коду програми.

*Макети об'єктів (англ. Mock Object) в об'єктно-орієнтованому програмуванні* – це об'єкти, що імітують поведінку справжніх об'єктів контрольованими способами, тобто, реалізують інтерфейси справжніх об'єктів, але не мають власної реальної функціональності.

## ВСТУП

У наш час багато підприємств вкладають великі кошти в розвиток своєї продукції й досить значні грошові ресурси вкладаються в рекламну кампанію. Тому для підприємства важливо знати, наскільки успішною є проведена рекламна кампанія.

Програмне забезпечення розроблене в цій дипломній роботі надає можливості для аналізу рекламної кампанії, а саме:

- аналіз якості проведення рекламної кампанії;
- аналіз динаміки продажів;
- аналіз популярності рекламної продукції;
- аналіз ефективності рекламних витрат;
- аналіз фінансових показників;
- динамічне оновлення даних, у разі зміни інформації;
- зручний інтерфейс;
- детальний перегляд графіків.

Якщо скласти всі вищезгадані компоненти в одне ціле, то отримаємо зручну та максимально прозору систему для аналізу рекламної кампанії.

Не зважаючи на те, що цей тип програми не є єдиним рішенням на ринку ПЗ, у цьому дипломному проєкті буде проаналізовано вже наявні рішення й розроблено останню версію продукту з урахуванням усіх мінусів наявних розроблень.

Основна мета дипломної роботи – розроблення програмного забезпечення, яке дає змогу аналізувати рекламну кампанію. Підставою для розроблення даного продукту стала проблема низької ефективності рекламної кампанії в підприємствах.

Це програмне забезпечення не потребує від користувача спеціальних фінансових та маркетингових знань та навичок програмування, що робить його доступним для кожного.

# 1. АНАЛІЗ НАЯВНИХ РІШЕНЬ ПОСТАВЛЕНОГО ЗАВДАННЯ

## 1.1 Загальні положення та аналіз предметної галузі

Аналіз рекламної кампанії – доволі складний і кропіткий процес. Ручний процес аналізу вимагає досить значних людських та часових ресурсів, тому автоматизація цього аналізу є актуальною проблемою.

На рис.1 можемо побачити дерево проблем поставленого завдання.

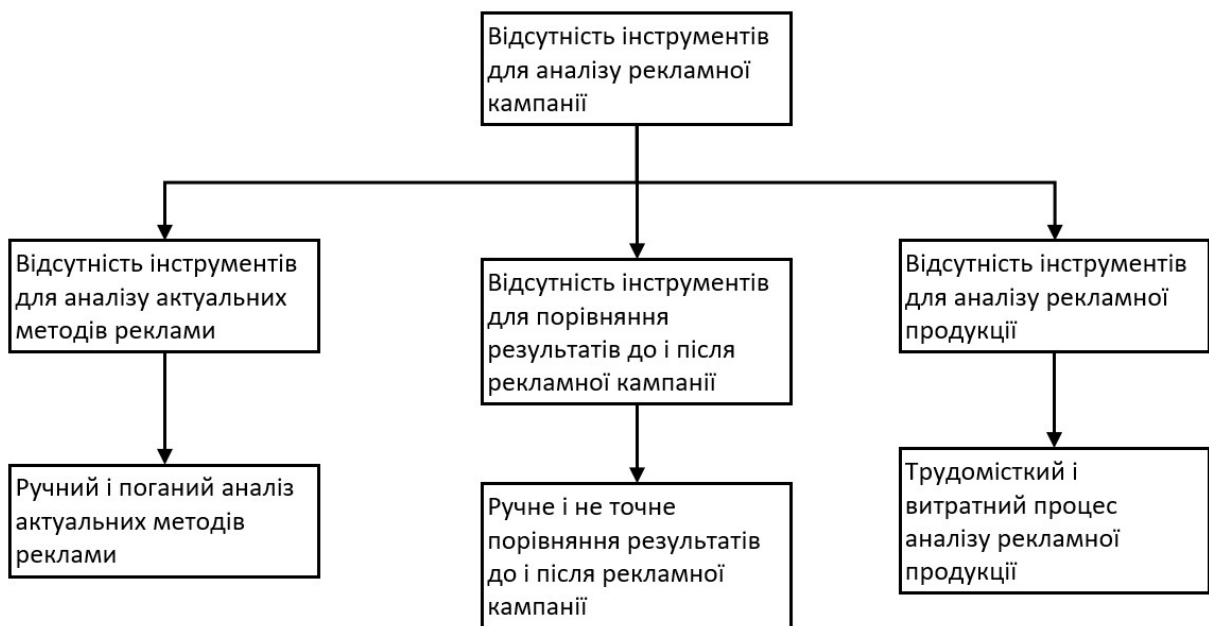


Рис. 1. Дерево проблем аналізу рекламної кампанії

Розглянемо більш детально предметну галузь для створення мобільного застосунку.

Аналіз рекламної кампанії має вирішувати такі основні завдання [1]:

- вимір (контроль) ефективності реклами;
- визначення характеристик носіїв реклами для вибору найбільш відповідних цілей просування;
- вимір основних показників медіаканалу;
- тестування сценарію (ідеї, концепції і т.д.), діяльності з просування товару.

Залежно від цілей і завдань рекламної кампанії в якості критеріїв її ефективності можуть бути різні показники.

Цілі рекламної кампанії можна розділити на дві великі групи:

- економічні;
- комунікативні.

Економічна ефективність реклами визначається за ступенем її впливу на розвиток товарообігу. Порівнюють також витрати на рекламу й той прибуток, який отримав рекламодавець внаслідок збільшення товарообігу під впливом реклами [2].

Комунікативна ефективність реклами характеризується числом охоплення споживачів, яскравістю і глибиною вражень, які рекламні засоби залишають в пам'яті людини, ступенем залучення уваги і т.д. [1].

Показниками економічної ефективності проведеної рекламної кампанії можуть служити:

- фактичне збільшення обсягу продажів;
- приріст числа нових клієнтів;
- збільшення частки на ринку.

До комунікативної ефективності можна віднести показники:

- упізнаваності торговельної марки;
- формування лояльності до торговельної марки.

Отже, аналіз ефективності реклами включає досить значну кількість показників. Розрахунок, який здійснюється «вручну», не влаштовує керівників підприємств та спеціалістів із маркетингу. Сучасний стан підприємств вимагає автоматизації процесу аналізу рекламної кампанії. Це значно зменшить втрати грошових коштів на підприємствах та надасть більш ефективно задовольняти потреби ринку.

## **1.2. Аналіз аналогічних застосунків**

Проаналізуємо програмні засоби, які мають на меті збір інформації по аналізу ефективності рекламної кампанії.

### *Програмне забезпечення «Google AdWords»*

Система контекстної реклами Google AdWords належить найбільшій пошуковій системі світу Google, яка активно бореться за лідерство на ринку надання рекламних послуг в інтернет мережі. Клієнти Google AdWords мають доступ до функціональної й технологічної платформи, що має:

- гнучкі налаштування таргетингової реклами;
- систему планування бюджетування рекламної кампанії;
- різні способи та кількість показу оголошень.

#### Переваги «Google AdWords»:

- можливість швидко залучити на сайт перших цільових відвідувачів;
- налаштування першої рекламної кампанії зазвичай займає не більш ніж годину;
- гнучкі налаштування бюджету внаслідок чого рекламодавці мають можливість встановлювати добовий бюджет і корегувати його залежно від результатів кампанії;
- вплив на ефективність рекламних оголошень внаслідок чого рекламні оголошення можуть показуватися на більш вигідних позиціях за більш високою ціною;
- можливість вибору майданчиків, на яких демонструються оголошення;
- зручний інтерфейс користувача;
- відомості про рекламні кліки;
- показники залученості аудиторії;
- показники конверсії;
- можливість налаштувати відстеження всіх можливих значущих конверсій, що здійснюються на сайті;

- аналізує інформацію по ключових словах, пошуковим запитам, оголошенням, пристроях, місцезнаходженням і будує звіти з детальною інформацією.

До переваг можна віднести також покупку, додавання товару в кошик, відправку заповненої форми заявки, дзвінок. Дані про вчинені переходи дають змогу проаналізувати ефективність рекламної кампанії й отримати точні дані по рентабельності, а також підрахувати PPC.

Недоліки «Google AdWords»:

Основний недолік цього рішення полягає в тому, що воно створене лише для аналізу рекламної кампанії в пошуковій системі Google.

### *Програмне забезпечення «Яндекс.Дірект»*

«Яндекс.Дірект» – рекламна система, за допомогою якої ви можете розміщувати контекстні оголошення на сторінках Яндекс.Пошуку й на партнерських сайтах рекламної мережі. Оголошення в Яндекс.Діректі показуються виключно тим людям, які вже зайняті пошуком схожих послуг і товарів на Яндексі та інших сайтах.

Переваги ПЗ:

- швидке залучення цільового трафіку: використовується контекстна реклама для отримання клієнтів із першого дня існування проєкту;
- ефективний принцип оплати: ви витрачаєте кошти тільки на переходи користувачів на сайт;
- гнучке таргетування аудиторії;
- просте управління рекламними кампаніями;
- лічильник метрик для обліку кліків і дій відвідувачів, збору даних про аудиторію сайту;
- оцінювання обсягу трафіку, формування портрета цільової аудиторії, відстеження цільових дій, аудит витрат на рекламу і виторг;

- загальна статистика, розбивка по днях, регіонах і майданчикам, на яких ви транслюєте рекламу: кількість показів і кліків, відсоток відмов, CTR за ключовими словами.

Недоліки ПЗ:

- це ПЗ розроблене більше для розміщення реклами, аніж для її аналізу;
- недоступне на території України.

### *Програмне забезпечення «Roistat»*

Roistat – це програмне забезпечення, що збирає дані з CRM системи, рекламних майданчиків і вашого сайту. За секунди будує звіти по 40+ бізнес-показниках.

Система відстежує витрати, рахує вартості кліка (PPC), ліда (CPL), оплати (CPO), окупність інвестицій (ROI) і приблизно 50 інших показників. Наприклад, прогнозований прибуток, повторні продажі, життєвий цикл клієнта (LTV).

Переваги ПЗ «Roistat»:

- має 9 інструментів для підвищення ефективності рекламних кампаній: зведений звіт, трекінг електронної пошти, трекінг дзвінків споживачів, «ловець» лідів, наскрізна й мультиканальна аналітика, автоуправління ставками, когортний аналіз, спліт-тестування;
- фільтрація сміттєвих лідів, відмов і повернень;
- гнучка система звітів, яка дає змогу за кілька секунд отримати дані за будь-який часовий проміжок по потрібному рекламному джерелу або конкретного менеджера, який займається обробкою заявки;
- під час спроби вийти з сайту, спливає вікно, внаслідок якого потенційний клієнт може передумати покидати сайт.



Недоліки ПЗ «Roistat»:

- громіздкий інтерфейс користувача;
- немає динамічного оновлення даних;
- висока ціна;
- складність у використанні.

У табл. 1 наведено аналіз переваг та недоліків вищезгаданих програмних систем та застосунків.

Таблиця 1

Аналіз переваг та недоліків програмних систем та застосунків

Показник	Google AdWords	Яндекс.Дірект	Roistat
Цінова доступність	✓	✓	✗
Швидкість	✓	✓	✗
Повнота функціональних можливостей	✗	✗	✓
Зручність	✓	✓	✗
Доступність	✓	✗	✓
Простота користування	✓	✓	✗
Гнучкість налаштування	✓	✓	✓
Ефективність використання	✗	✗	✓

### 1.3. Актуальність розроблення

На сьогодні, усі підприємства потребують якісного й детального аналізу рекламної кампанії. Вищеописані системи мають недоліки та аналізують рекламу лише в інтернет мережі й тому постає питання в створенні нового додатку для аналізу рекламної кампанії.

Як показують дослідження, більшість ринку належить мобільним застосункам, які працюють на операційних системах Android та iOS. Приблизно 70% людей використовують свій смартфон від 5 до 7 годин у день. До того ж смартфон зручний у використанні, недорогий, не потребує додаткових аксесуарів та не має бути потужним для того щоби підтримувати найактуальніші застосунки. Також мобільні застосунки мають найкращу перевагу – це їхня ціна, завдяки чому користуватися ними можуть не лише великі підприємства, але й малі, а також повністю нові проєкти, наприклад, стартапи. Тому було прийнято рішення реалізувати програмне забезпечення для аналізу рекламної кампанії у вигляді iOS-застосунку.

#### **1.4. Вимоги до програмного забезпечення**

На основі аналізу розглянутих наявних аналогів та попереднього опитування цільової аудиторії було розроблено відповідні загальні вимоги до системи.

Це програмне забезпечення має відповідати таким вимогам та мати:

- забезпечувати функціонування застосунку одночасно більше ніж двох користувачів;
- сервер для збору даних з CRM-системи, з бази даних чи напряму із сайту;
- можливість вводу, редагування чи додавання необхідної інформації вручну;
- підтримувати відповідне шифрування даних задля підтримання інформаційної безпеки;
- забезпечення різних рівнів доступу та прав для користувачів – адміністратор, звичайний користувач;
- ПЗ має мати алгоритм щодо розподілу рекламних бюджетів;
- надавати всю необхідну інформацію впродовж рекламної кампанії: кількість клієнтів, витрати, кількість проданих товарів, показники рекламних бюджетів тощо;

- особистий кабінет користувача;
- модуль аналітики рекламних каналів;
- модуль для генерування таблиць;
- модуль для аналізу рекламної продукції – аналізує динаміку популярності рекламної продукції в соціальних мережах та інтернеті;
- модуль для оплати сервісу;
- модуль для побудови графіків;
- модуль для генерування звітів;
- модуль для динамічного оновлення інформації;
- модуль, який буде відповідати за безпеку обміну даних між сервером та клієнтом;
- модуль для збереження даних у БД для подальшого використання.

Розроблене ПЗ має відповідати таким нефункціональним вимогам:

- підтримання останніх версій iOS;
- максимально простий, зручний для користувача та адаптивний інтерфейс;
- модуль для зміни валюти;
- модуль для зміни мови.

Вимоги до якості розроблюваного ПЗ:

- відповідне шифрування даних для забезпечення конфіденційності даних проєкту;
- швидкісні характеристики;
- безперервний доступ для користувачів.

Вимоги до безпеки ПЗ:

- захист від SQL-ін'єкцій;
- перевірка введених даних;
- шифрування даних, що зберігаються.

## **1.5. Висновки до розділу**

Аналіз наявного програмного забезпечення для аналізу рекламних кампаній показує, що таке програмне забезпечення в основному спрямоване на рекламну кампанію в інтернет мережі. Зі свого боку комплексне програмне забезпечення «Roistat» є дуже громіздким та дорогим. Отже, на сьогодні немає програмного забезпечення, яке дає змогу ефективно аналізувати рекламну компанію для малого та середнього бізнесу.

Проаналізувавши актуальність проблеми та ринок застосунків, було прийнято рішення, що розроблюване програмне забезпечення має бути представлене у вигляді iOS-застосунку, забезпечувати ефективний аналіз рекламної кампанії та надавати всі необхідні інструменти.

Програмне забезпечення «АСА» має бути зручним та простим, що є головною його перевагою. Також воно має бути недорогим, що надасть змогу використовувати його малому та середньому бізнесу, в тому числі стартапам.

Також було визначено вимоги до розроблюваного програмного забезпечення: воно має збирати дані з різних джерел, надавати всю необхідну інформацію впродовж рекламної кампанії, аналізувати рекламну продукцію, а також підтримувати останні версії iOS та мати зручний інтерфейс.

## **2. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ**

### **2.1. Обґрунтування вибору ОС мобільного застосунку**

На ринку є багато мобільних ОС та лише дві з них є актуальними на сьогодні. Проаналізуємо, яку краще вибрати для реалізації поставленого завдання.

#### *ОС «Android»*

Android – мобільна операційна система для смартфонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Найбільше всього смартфонів саме на цій ОС. До того ж, саме Android пристроям належить більшість трафіку в інтернеті [3].

Основною перевагою Android є те, що вона, на відміну від iOS, використовується не тільки на пристроях компанії Google, але й на пристроях інших виробників, завдяки чому користувач має широкий вибір. Проте, для розробників це великий мінус, оскільки необхідно підтримувати роботу застосунків на всіх наявних пристроях і дуже складно створити універсальний застосунок, який буде працювати однаково на всіх наявних пристроях на ОС Android.

Для розповсюдження застосунків компанія Google використовує власний застосунок Google Play. Google Play – магазин застосунків від компанії Google, за допомогою якого інші компанії мають змогу розповсюджувати свої застосунки, у тому числі ігри, фільми, книги та музику, як безкоштовно так і платно. Обліковий запис розробника, який дає можливість публікувати застосунки, коштує 25 доларів. Платні додатки можуть публікувати розробники не з усіх країн [4].

Є багато програмних засобів для створення мобільних застосунків для ОС Android. Основним програмним засобом є Android Studio від Google. Android Studio в порівнянні з іншими програмними засобами має декілька переваг: вбудовані засоби розроблення, можливість написання мобільних застосунків на різних мовах програмування, підтримання розробників. Проте, ця IDE має значний недолік - повільна симуляція застосунків. Тому для того, щоби протестувати застосунок на всіх наявних пристроях потрібно їх мати, або мати потужну систему для роботи симуляції [5].

### *ОС «iOS»*

iOS – мобільна ОС розроблена компанією Apple. Вона використовується лише на пристроях компанії Apple, на відміну від Android. iOS займає друге місце за кількістю пристроїв у світі.

Для поширення застосунків компанія Apple використовує фірмовий магазин застосунків App Store. Цей магазин був створений раніше, ніж його аналог на операційній системі Android. В App Store розміщено понад 2 мільйонів застосунків для iOS, які були завантажені понад 15 мільярдів разів [6].

Для розроблення мобільних застосунків на ОС iOS використовується IDE Xcode, створена компанією Apple. На iOS можна створювати застосунки лише на двох мовах програмування – Objective C та Swift. На відміну від Android Studio, Xcode має потужний вбудований симулятор, з допомогою якого не обов'язково мати всі наявні пристрої компанії Apple та можна переглянути вигляд та роботу свого програмного забезпечення на всіх наявних пристроях Apple та версіях iOS [7].

На відміну від Android, який не має зв'язків з іншими системами чи пристроями, мобільна операційна система компанії Apple є частиною великої екосистеми, завдяки чому різні пристрої компанії можуть

синхронізуватися та обмінюватися інформацією, що значно полегшує життя користувачам та процес розроблення застосунків програмістам.

Оскільки iOS використовується лише на пристроях вироблених власноруч компанією Apple, то смартфони на базі iOS будуть підтримуватися розробниками роками. Це полегшує процес розроблення програмного забезпечення та зменшує кількість можливих помилок у роботі з ним.

Основною перевагою iOS над Android є безпека. На відміну від Android, де дуже розвинене «піратство» й де без сторонньої допомоги можна з легкістю й безкоштовно завантажити мобільний застосунок із будь-якого ресурсу, а не тільки з офіційного магазину, на iOS застосунки можна завантажувати лише в App Store із чого виходить, що її користувачі є більш платоспроможними. Усі операційні системи, сервіси та пристрої від компанії Apple захищені на найвищому рівні, через що витік критичних даних майже неможливий і саме тому здебільшого користувачі обирають цю компанію.

### *Обрана ОС*

Переглянувши наявні варіанти ОС для реалізації мобільного застосунку була вибрана мобільна ОС iOS, тому що вона безпечніша, підтримується компанією Apple як на програмному рівні так і на апаратному та входить до великої екосистеми Apple, завдяки чому процес розроблення програмного забезпечення набагато легший.

## **2.2. Обґрунтування вибору мови програмування**

### *Мова програмування Swift*

Swift – багатопарадигмова компільована мова програмування, для розроблення застосунків для ОС iOS, tvOS, watchOS та macOS. Swift був

розроблений для того, щоби замінити мову Objective C і бути стійкішим до помилкового коду, а також бути простішим у написанні коду[8].

Код застосунків написаних мовою програмування Swift легше зрозуміти в порівнянні з Objective-C. Swift використовує основні принципи сучасних мов програмування. Також Swift має унікальну особливість Swift Playgrounds з допомогою якої проміжні результати програми можна побачити, не компілюючи її. До того ж, мова програмування Swift дуже швидка. За заявою Apple, код Swift компілюється в 1.3 рази швидше коду на Objective-C.

Swift – мова програмування схожа на програмну мову C. Але оскільки це одна з найновіших мов програмування, вона має низку нововведень, що полегшують процес написання коду. Також в Swift процес збирання сміття автоматизований, внаслідок чого можна позбутися від використання збирача сміття й необхідності програмістом слідкувати за цим.

Завдяки гнучкості мови програмування Swift, застосунки можуть бути написані одночасно на декількох мовах програмування.

Переваги Swift:

- швидкість;
- зручність;
- безпека;
- простий та швидкий процес навчання;
- зручна IDE Xcode.

Недоліки Swift:

- нестабільність через її новизну;
- мала кількість бібліотек;
- розроблена лише для техніки Apple.

*Мова програмування Objective-C*

Objective-C – високорівнева об'єктно-орієнтована мова програмування загального призначення, яку створила компанія Apple на



початку 1980-х років для операційної системи NeXTSTEP, від якої пішли macOS і iOS. Більшість мобільних застосунків на ОС iOS, так само як і сама iOS, написані мовою програмування Objective-C. Упродовж довгого періоду часу розробляти мобільні застосунки на техніку Apple можна було тільки на цій мові програмування, доки в 2014 році не була представлена мова програмування Swift, що прийшла їй на заміну [9].

Ця мова програмування вирішувала багато тогочасних проблем і на той момент це була найкраща мова програмування. Проте, на сьогодні ця мова програмування є застарілою й не вирішує сьогоденні проблеми в програмуванні, наприклад, такі, як збирання сміття.

Компанія Apple роками намагалася покращити цю мову програмування, проте, розробляючи новий функціонал для виправлення старих помилок, з'являлися нові більш критичні помилки. З часом синтаксис мови став більш зрозумілим для спрощення вивчення мови, проте не було вирішено основної проблеми цієї мови – а саме збирання сміття, тому що, як би Apple не намагалася покращити роботу цього механізму, це не допомагало й не вирішувало поставленого завдання.

#### Переваги Objective-C:

- більшість застосунків для продукції Apple написано саме на цій мові;
- не має строгої типізації;
- сумісна з мовою програмування C, що дає змогу використовувати її функціонал.

#### Недоліки Objective-C:

- нечитабельність;
- повільна швидкість програмного коду;
- незручність;
- застарілість – більше не оновлюється та не підтримується;
- буде повністю замінена мовою Swift.

### *Обрана мова програмування*

Отже, розглянувши наявні варіанти створення мобільних застосунків на операційну систему iOS, було прийнято рішення розробляти програмне забезпечення на мові програмування Swift, тому що вона безпечна, зручна та проста. Проте, оскільки вона підходить тільки для розроблення програмного забезпечення для техніки Apple та через її новизну має невелику кількість бібліотек, сервер буде реалізований на іншій мові програмування.

## **2.3. Обґрунтування вибору мови програмування для серверу**

### *Мова програмування Javascript*

JavaScript (JS) – динамічна, об'єктно-орієнтована мова програмування. В основному використовується для створення вебсайтів та серверів. Найпопулярніша мова програмування [10].

Ця мова програмування володіє великим спектром функцій, серед яких є динамічна типізація, автоматичне керування пам'яттю, наявність анонімних функцій, наявність засобів тестування, велика кількість бібліотек тощо.

Не зважаючи на те, що JS класифікують, як прототипну мову програмування, вона також підтримує інші парадигми, наприклад, імперативну.

Підтримується всіма компаніями та розробниками, внаслідок чого розроблення сайтів і серверів на цій мові програмуванні займає небагато часу.

Переваги JavaScript:

- мова постійно оновлюється;
- простий синтаксис;
- можливість запуску застосунків на сервері;
- кросплатформеність;

- широкий вибір бібліотек.

Недоліки JavaScript:

- недостатній рівень безпеки;
- наявність дрібних помилок;
- складна парадигма.

### *Мова програмування PHP*

PHP – це мова програмування в основі якої лежать сценарії і яка часто використовується для розроблення вебзастосунків. Сценарій в PHP – це програма, яка знаходиться на стороні сервера і виконується у відповідь на запит від браузера. Виконання сценарію можна також назвати інтерпретацією і з чого слідує, що мова програмування PHP є інтерпретатором [11].

Мова програмування PHP є другою за популярністю у світі після Javascript і тому може використовуватися на будь-якій операційній системі та у будь-якому браузері.

Зазвичай сторінки в браузері є статичними, проте це не завжди гарно. Наприклад, сторінка показує курс валют, який постійно оновлюється і для того, щоб побачити зміни потрібно оновити сторінку. Цього можна уникнути за допомогою мови програмування PHP, бо він прийме запит і виконавши його відобразить динамічну сторінку на якій постійно будуть оновлюватися дані без необхідності оновлення сторінки.

Мова програмування PHP не є простою й потребує певних знань для умілого володіння нею, тому що вона була розроблена аж в 1995 році. Завдяки тому, що на ринку з'являлися інші мови програмування з часом мова PHP позичала деякі можливості в них, завдяки чому стала набагато простішою, проте все ще відстає від своїх аналогів, наприклад, Javascript.

Переваги PHP:

- сумісність з усіма наявними базами даних;
- відкритий вихідний код;

- безплатна ліцензія;
- велика кількість бібліотек та модулів;
- висока швидкість обробки запитів.

Недоліки РНР:

- складність;
- рідко оновлюється;
- погана графічна складова.

### *Мова програмування Python*

Python – це мова програмування за допомогою якої можна розробляти будь-які програмні забезпечення в тому числі вебсайти, сервери, мобільні застосунки тощо. Python налічує безмежну кількість бібліотек, унаслідок чого може використовуватися для вирішення будь-яких завдань, наприклад, збір і аналіз інформації на будь-яких операційних системах та пристроях. Код у Python розділений на модулі, унаслідок чого спрощується тестування програмного забезпечення. Мова програмування Python використовує такі парадигми програмування: об'єктно-орієнтована, функціональна, процедурна та аспектно-орієнтована [12].

Python – це найпростіша мова програмування у світі навчитися користуватися якою можна менше ніж за тиждень. Проте, це є як її перевагою так і недоліком. Тому що, за рахунок своєї простоти в ній відсутня більшість функціоналу сучасних мов програмування внаслідок чого вона є повільною й написані програмні забезпечення на ній мають багать помилок у тому числі критичних, унаслідок чого її не рекомендують використовувати для розроблення програмного забезпечення.

Ця мова програмування, на відміну від багатьох інших може використовуватися одночасно з іншими мовами програмування та має спеціальний інтерфейс та функції для цього.

До того ж, мова програмування Python має зручну IDE для роботи з нею – PyCharm. Вона налічує в собі зручний інтерфейс, функціональні

можливості для тестування програмного забезпечення, а також зручний спосіб додавання сторонніх модулів чи функцій до програмного забезпечення.

Завдяки тому, що основним принципом цієї мови програмування є модульність, дуже просто й легко використовувати функції одного модуля в іншому.

За замовчуванням мова програмування Python не підтримує асинхронне програмування. І хоча за рахунок сторонніх бібліотек і модулів можна досягти асинхронного програмування цей процес є дуже важким і потребує багато часу, не дивлячись на простоту мови програмування Python.

Переваги Python:

- простота;
- кросплатформеність;
- велика кількість бібліотек та модулів;
- найкращі функції для розв'язання математичних задач;
- відкритий вихідний код.

Недоліки Python:

- велика кількість помилок;
- повільна швидкість компіляції програм;
- складна реалізація асинхронного програмування.

*Обрана мова програмування для написання серверу*

Отже, проаналізувавши мови програмування для створення серверу, було прийнято рішення розробляти сервер на мові програмування Javascript, тому що вона популярна, зручна та проста.

## 2.4. Обґрунтування вибору системи керування базами даних

### *СКБД MySQL*

MySQL є найпопулярнішою базою даних із відкритим вихідним кодом. Із самого початку вона була створена, як альтернатива платним системам. Проте, вона є безплатною тільки у випадку, коли програмне забезпечення, що її використовує, знаходиться у відкритому доступі. В іншому випадку необхідно купувати комерційну ліцензію [13].

Дана СКБД має великий функціонал, завдяки якому можна працювати з різними типами таблиць.

Завдяки відкритому вихідному коду, користувачі можуть його змінювати, щоби він краще відповідав їхнім потребам. У цьому полягає гнучкість системи.

Вона має простий інтерфейс та зручно обробляє великі обсяги даних, за допомогою пакетних команд. Система надійна й не потребує дорогого та потужного сервера для роботи з нею.

#### Переваги СКБД MySQL:

- є безплатна версія, що підходить для невеликих проєктів;
- зрозуміла документація;
- відкритий вихідний код;
- легка у використанні;
- зручний інтерфейс;
- сумісна з іншими базами даних.

#### Недоліки СКБД MySQL:

- більшість завдань не автоматизовані, що потребує людського втручання;
- немає вбудованої підтримки XML або OLAP;
- безплатна версія доступна лише програмним забезпеченням із відкритим вихідним кодом.

### *СКБД Realm*

Realm – це об’єктна база даних із відкритим вихідним кодом розроблена виключно для мобільних застосунків. Ця база даних не передбачає безплатної ліцензії й коштує дуже дорого, тому що розроблена виключно для великих підприємств [14].

Взаємовідношення між об’єктами в цій СКБД відбувається завдяки посиланням. Кожне нове посилання створює зворотне посилання, у результаті чого, пройшовши ланцюгом посилань, ми дійдемо до базової таблиці. В результаті чого, будь-які зміни в будь-якій таблиці впливають на всі інші таблиці.

#### Переваги СКБД Realm:

- велика швидкість обробки запитів;
- простота у використанні;
- легкість інтеграції.

#### Недоліки СКБД Realm:

- розроблена виключно для мобільних пристроїв;
- не передбачає безплатної ліцензії;
- нестабільність через її новизну.

### *СКБД MongoDB*

MongoDB – документо-орієнтована безплатна база даних з відкритим вихідним кодом. Вона є функціональною і зручно формує запити, а також працює швидко як із малим обсягом даних так і з великим [15].

Ця база даних оперує даними у форматі ключ-значення та зберігає об’єкти у JSON-форматі, внаслідок чого вони є зрозумілими для людей і пошук у базі даних працює так само, як пошук на сайті. Постійно оновлюється та має великий спектр бібліотек, унаслідок чого має можливість використовуватися на будь-якій мові програмування. Має зручний застосунок для роботи з даними, за допомогою якого MongoDB використовується без додаткових знань.

### Переваги СКБД MongoDB:

- зберігає дані у JSON-форматі, за допомогою чого вони є читабельними;
- ефективне зберігання великих даних, наприклад, фото та відео;
- підтримка відмовостійкості й масштабованості;
- зручний застосунок для роботи з базою даних;
- динамічні запити;
- фіксування будь-яких дій пов'язаних із базою даних;
- швидке оновлення даних;
- гнучкість;
- легкість у використанні.

### Недоліки СКБД MongoDB:

- дані не є структурованими;
- немає жорсткої типізації даних;
- немає ізоляції – будь-які дані, що зчитуються одним користувачем, можуть у той самий час змінюватися іншим.

### *Обрана СКБД*

Проаналізувавши найкращі варіанти з усіх наявних баз даних було прийнято рішення обрати СКБД MongoDB, тому що вона є повністю безплатна, має зручний застосунок для роботи з нею та підтримується всіма наявними мовами програмування.

## **2.5. Шаблон проєктування MVC**

Model-View-Controller (MVC) – шаблон проєктування, який використовується для полегшення розроблення програмного забезпечення в тому числі мобільних застосунків. Основною ідеєю цього шаблону є поділ програмного забезпечення на три рівнозначні частини (рис. 2). Це модель, інтерфейс та система керування. Як правило, цей шаблон застосовується для



розроблення мобільних застосунків. Підтримується всіма наявними мовами програмування [16].

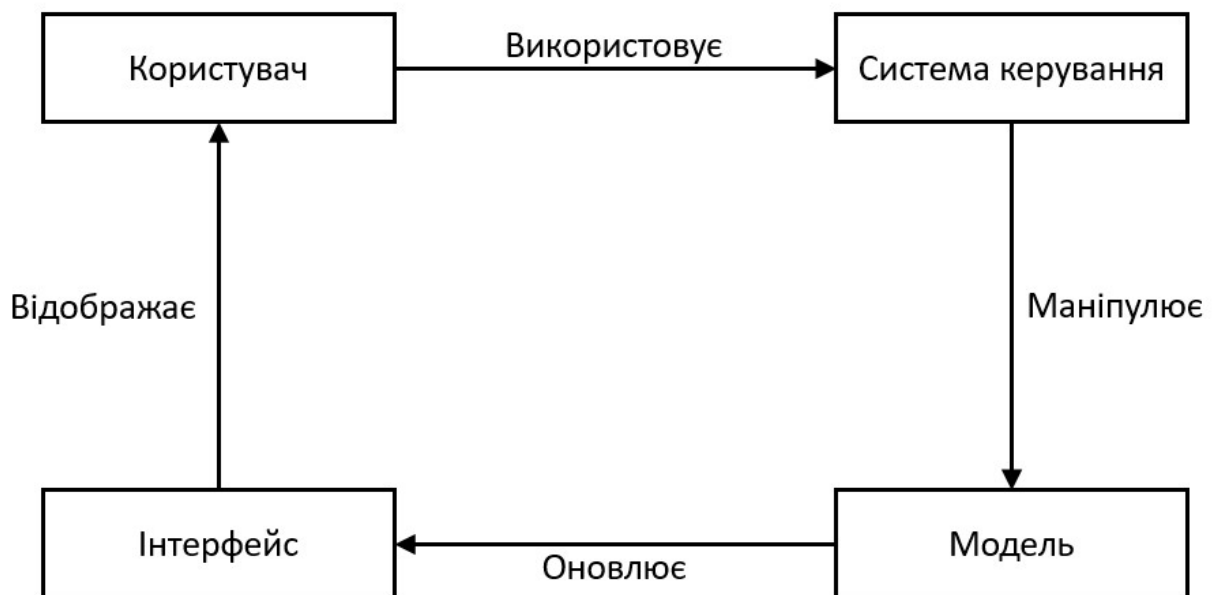


Рис. 2. Діаграма взаємодії між компонентами шаблону MVC

Основним завданням цього шаблону є полегшення сприйняття громіздкого програмного забезпечення за рахунок відокремлення інтерфейсу та моделі.

Принцип роботи цього шаблону полягає в тому, що інтерфейс приймає вхідні дані та команди та доставляє їх до моделі, яка в свою чергу оновлює інтерфейс для відображення його користувачеві. У різних мовах програмування цей процес реалізований по-різному й не виключає можливості контролера посилати команди як до інтерфейсу так і до моделі.

Модель – найважливіша частина шаблону проєктування MVC, яка відповідає за збереження та маніпуляцію над даними. Дуже часто використовується для запису та зчитування даних із БД. Не повинна залежати від інших компонентів програмного забезпечення. Дані можуть зберігатися в різних форматах та змінюватися під час роботи з програмним забезпеченням.

Ознаки моделі:

- захист даних;
- серце програмного забезпечення;
- доступ до БД.

Інтерфейс відповідає лише за відображення даних отриманих від моделі й у жодному разі не може їх змінювати. Процес відображення може відбуватися за рахунок графіків, таблиць, текстових полів тощо. Як правило, інтерфейс не впливає на інші компоненти програмного забезпечення.

Контролер – це керівник програмного забезпечення. Його основною задачею є доставлення команд до моделі. Як правило, ці команди передає користувач за рахунок вводу даних та натисканню кнопок. Проте, іноді контролер використовується для доставлення команд до інтерфейсу, наприклад, якщо користувач хоче змінити зовнішній вигляд програмного забезпечення..

Завдяки MVC спрощується сприйняття програмного забезпечення та його подальший розвиток.

## **2.6. Висновки до розділу**

У цьому розділі було розглянуто та проаналізовано всі можливі шляхи реалізації мобільного застосунку. У результаті була обрана операційна система, мова програмування та СКБД для реалізації програмного забезпечення.

Операційною системою була обрана ОС від компанії Apple – iOS, тому що вона безпечніша та зручніша, а також входить до великої екосистеми компанії Apple, унаслідок чого її клієнти мають високу платоспроможність і програмне забезпечення буде безпечним і підтримуватися компанією Apple.

Мовою програмування була обрана мова Swift, тому що вона новіша, зручніша й легша, а також підтримується розробниками. Для реалізації

сервера була обрана мова Javascript, оскільки мова Swift розроблена виключно для техніки Apple.

СКБД була обрана MongoDB через її зручність, простоту та швидкість обробки даних та операцій із ними.

### **3. РОЗРОБЛЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ**

#### **3.1. Загальний вигляд мобільного застосунку**

Мобільний застосунок був розроблений із виконанням вимог описаних у підрозділі 1.4.

Неавторизований користувач має доступ лише до сторінки реєстрації, на якій йому необхідно створити новий акаунт чи зайти вже наявний. Для створення нового акаунта користувач вводить логін та пароль. Після реєстрації користувач має можливість додати додаткову інформацію про себе в особистому кабінеті.

Є два типи користувачів: «Admin», який має можливість виконувати всі доступні функції в застосунку, а також переглядати дії інших користувачів та змінювати роль користувача і «Default», який має можливість лише переглядати файли, до яких у нього є доступ.

Після авторизації в системі користувач потрапляє на сторінку створення чи вибору файлу аналізу рекламної кампанії, на якій необхідно вказати джерело інформації, назву файлу та, за наявності, обрати звіт по аналізованій у минулому рекламній кампанії.

Авторизовані користувачі мають доступ до:

- «Особистий кабінет» – на цій сторінці користувач має змогу переглядати чи змінювати особисту інформацію, а саме логін, пароль, ім'я, дату народження та назву підприємства;
- «Головна сторінка» – на цій сторінці відображається вся необхідна інформація для аналізу рекламної кампанії;
- «Аналіз продукції» – на цій сторінці відображається зріст популярності рекламної продукції в соціальних мережах та в інтернеті;
- «Звіт» – ця сторінка необхідна для завершення аналізу рекламної кампанії та генерації звіту;

- «Журнал дій» – ця сторінка доступна лише користувачам із роллю «Admin» та відображає дії всіх користувачів;
- «Графік» – на цій сторінці можна детально розглянути з додатковими можливостями будь-який графік;
- «Додаткова інформація» – сторінка з інформацією про застосунок.

### 3.2. Аналіз функціональних вимог до мобільного застосунку

На першому етапі процесу розроблення мобільного застосунку насамперед необхідно сформулювати та проаналізувати його функціональні вимоги. Функціональні вимоги – це ті вимоги, що описують як саме має працювати мобільний застосунок та що він має робити [17]. Детальний аналіз функціональних вимог наведено в табл.2.

Таблиця 2

Функціональні вимоги до мобільного застосунку для аналізу рекламної кампанії

Номер	Назва	Пріоритет
1	Реєстрація в застосунку	1
2	Авторизація в застосунку	1
3	Детальний перегляд графіків	2
4	Створення файлу для аналізу рекламної кампанії	1
5	Створення звіту по завершенні рекламної кампанії	1
6	Редагування особистої інформації	3
7	Перегляд дій користувачів	3
8	Перегляд основної інформації під час рекламної кампанії	1

9	Перегляд зросту популярності рекламної продукції	1
10	Можливість зворотного зв'язку з розробником	4
11	Можливість перегляду інформації про мобільний застосунок	4

Переглянувши і проаналізувавши вимоги до мобільного застосунку для аналізу рекламної кампанії було визначено, які саме можливості має мобільний застосунок і які з них є найважливішими й потребують реалізації найперше.

### 3.3. Архітектура мобільного застосунку

#### *Опис системних файлів та бібліотек*

У мобільному застосунку наявні такі системні файли та бібліотеки:

- Info.plist – файл за допомогою якого відбувається доступ до всіх можливих властивостей програми і зміни їхніх параметрів (наприклад, можливі орієнтації мобільного застосунку) [18];
- AppDelegate.swift – основний клас програми, який керує діями інших класів, а також самим застосунком (що, як і коли запускати), що створюється за замовчуванням і немає змінюватися програмістом [19];
- Main.storyboard – файл, у якому наглядно відображаються всі можливі екрани мобільного застосунку, на яких розташовані елементи інтерфейсу (кнопки, текстові поля, графіки тощо) та зв'язки між екранами [20];
- Assets.xcassets – тека, у яку завантажені всі зображення, що використовуються в мобільному застосунку, за допомогою якої

спрощується доступ до ресурсів, бо за допомогою назви одного файлу можна знайти його відображення для різних орієнтацій чи різних за величиною пристроїв [21];

- `LaunchScreen.storyboard` – стартовий екран, який відображається впродовж декількох мілісекунд після запуску користувачем мобільного застосунку, що необхідно для покращення сприйняття швидкості мобільного застосунку [22];
- `Charts` – бібліотека для візуалізації даних у вигляді графіків;
- `RNCryptor` – бібліотека для шифрування даних;
- `iOSDropDown` – бібліотека для реалізації випадаючих списків.

### *База даних*

База даних для збереження та отримання інформації налічує 5 таблиць:

- таблицю користувачів;
- таблицю файлів аналізу рекламної кампанії;
- таблицю рекламних каналів;
- таблицю файлів аналізу рекламної продукції;
- таблицю звітів;
- таблицю дій користувачів.

Таблиця користувачів має наступні поля:

- логін – текстове поле для зберігання логіну користувача;
- пароль – текстове поле для зберігання паролю користувача;
- ім'я – текстове поле для зберігання імені користувача;
- роль – текстове поле для зберігання ролі користувача;
- дата народження – поле з форматом дати, що зберігає дату народження користувача;
- назва підприємства – поле, що зберігає назву підприємства користувача;

- назви створених файлів аналізу рекламної кампанії – масив, що зберігає назви створених файлів аналізу рекламної кампанії користувача;
- назви створених файлів рекламної продукції – масив, що зберігає назви створених файлів аналізу рекламної продукції користувача;
- назви створених звітів – масив, що зберігає назви створених звітів користувача.

Таблиця файлів аналізу рекламної кампанії має наступні поля:

- назва – текстове поле, що зберігає назву файлу рекламної кампанії;
- логін користувача – текстове поле, що зберігає логін користувач, що створив файл;
- дата початку – поле з форматом дати, що зберігає дату початку рекламної кампанії;
- бюджет – цифрове поле, що зберігає бюджет рекламної кампанії;
- поточна дата – поле з форматом дати, що зберігає поточну дату;
- кількість проданих товарів – цифрове поле, що зберігає кількість проданих товарів за поточну дату;
- кількість нових клієнтів – цифрове поле, що зберігає кількість нових клієнтів за поточну дату;
- прибуток – цифрове поле, що зберігає прибуток за поточну дату;
- тип джерела – текстове поле, що зберігає тип джерела;
- посилання на джерело – текстове поле, що зберігає посилання на джерело;
- ROI – цифрове поле, що зберігає ROI за поточну дату;
- PPC – цифрове поле, що зберігає PPC за поточну дату;
- CPO – цифрове поле, що зберігає CPO за поточну дату;
- CPL – цифрове поле, що зберігає CPL за поточну дату.



Таблиця рекламних каналів має наступні поля:

- назва файлу – текстове поле, що зберігає назву файлу рекламної кампанії, до якого ця таблиця відноситься;
- назва рекламного каналу – текстове поле, що зберігає назву рекламного каналу;
- поточна дата – поле з форматом дати, що зберігає поточну дату;
- кількість проданих товарів – цифрове поле, що зберігає кількість проданих товарів за поточну дату по заданому рекламному каналу;
- кількість нових клієнтів – цифрове поле, що зберігає кількість нових клієнтів за поточну дату по заданому рекламному каналу;
- прибуток – цифрове поле, що зберігає прибуток за поточну дату по заданому рекламному каналу;
- ROI – цифрове поле, що зберігає ROI за поточну дату по заданому рекламному каналу;
- PPC – цифрове поле, що зберігає PPC за поточну дату по заданому рекламному каналу;
- CPO – цифрове поле, що зберігає CPO за поточну дату по заданому рекламному каналу;
- CPL – цифрове поле, що зберігає CPL за поточну дату по заданому рекламному каналу.

Таблиця файлів аналізу рекламної продукції має наступні поля:

- назва файлу – текстове поле, що зберігає назву файлу рекламної кампанії, до якого ця таблиця відноситься;
- назва джерела – текстове поле, що зберігає назву джерела;
- поточна дата – поле з форматом дати, що зберігає поточну дату;
- кількість згадок – цифрове поле, що зберігає кількість згадок за поточну дату по заданому джерелу.

Таблиця звітів має наступні поля:

- назва файлу – текстове поле, що зберігає назву файлу рекламної кампанії, до якого ця таблиця відноситься;
- дата початку – поле з форматом дати, що зберігає дату початку рекламної кампанії;
- дата завершення – поле з форматом дати, що зберігає дату завершення рекламної кампанії;
- масив рекламних каналів – масив елементи якого зберігають назву рекламного каналу та його показники;
- бюджет – цифрове поле, що зберігає бюджет рекламної кампанії;
- ROI – цифрове поле, що зберігає загальний ROI;
- PPC – цифрове поле, що зберігає загальний PPC;
- CPO – цифрове поле, що зберігає загальний CPO;
- CPL – цифрове поле, що зберігає загальний CPL.

Таблиця дій користувачів має наступні поля:

- назва файлу – текстове поле, що зберігає назву файлу рекламної кампанії, до якого ця таблиця відноситься;
- логін – текстове поле, що зберігає логін користувача, який виконав дію над файлом;
- дата – поле з форматом дати, що зберігає дату виконання дії;
- дія – текстове поле, що зберігає дію користувача над файлом.

### **3.4. Структура мобільного застосунку**

Мобільний застосунок був реалізований за допомогою архітектурного шаблону MVC. За замовчуванням усі мобільні застосунки написані на мові програмування Swift використовують саме цей шаблон.

Ідеологією цього шаблону є поділ програмного забезпечення на три рівнозначні частини: модель, інтерфейс та частина керування. Модель відповідає за структуру даних, їхнє коректне зберігання та оновлення на

інтерфейсі. Інтерфейс відповідає за відображення даних користувачеві. Частина керування необхідна для обробки та доставлення вхідних даних та команд до моделі та інтерфейсу, а також їхнім маніпулюванням [16].

На мові програмування Swift за модель відповідають класи, за інтерфейс відповідають екрани (View), що містяться у файлі Main.storyboard та за керування відповідають контролери екранів (ViewController).

### *Класи*

У мобільному застосунку наявні такі класи:

- User;
- AdvertisementAnalyze;
- ProductAnalyze;
- AdvertisementAnalyzeIndexes;
- AdvertisementAnalyzeSourceIndexes;
- ProductAnalyzeSourceIndexes;
- Report;
- ActionLog.

User – цей клас відповідає за створення нового користувача та збереження інформації про нього. У ньому наявні такі поля: логін, пароль, ім'я, роль, дата народження, назва підприємства, назви створених файлів аналізу рекламної кампанії, рекламної продукції та звітів. За замовчуванням усі поля, окрім логіну та паролю пусті. Об'єкт цього класу створюється під час реєстрації нового користувача. Це головний клас системи, до якого прив'язані інші класи.

AdvertisementAnalyze – цей клас відповідає за створення нового файлу аналізу рекламної кампанії та його збереження. У ньому наявні такі поля: назва, масив показників для аналізу рекламної кампанії, масив рекламних каналів із їхніми показниками, дата створення, бюджет. Усі поля мають бути заповнені для збереження. За замовчуванням це робиться автоматично.

ProductAnalyze – цей клас відповідає за створення та збереження файлу аналізу рекламної продукції. У ньому наявні такі поля: масив джерел із їхніми показниками.

AdvertisementAnalyzeIndexes – цей клас відповідає за збереження об'єктів масиву показників для аналізу рекламної кампанії. У ньому наявні такі поля: дата, кількість клієнтів, прибуток, кількість проданих товарів.

AdvertisementAnalyzeSourceIndexes – цей клас відповідає за збереження об'єктів масиву показників рекламних каналів. У ньому наявні такі поля: дата, назва каналу, бюджет, кількість клієнтів, прибуток, кількість проданих товарів.

ProductAnalyzeSourceIndexes – цей клас відповідає за збереження об'єктів масиву джерел для аналізу популярності рекламної продукції з їхніми показниками. У ньому наявні такі поля: дата, назва джерела, кількість запитів.

Report – цей клас відповідає за створення та збереження звіту по завершенні рекламної кампанії. У ньому наявні такі поля: назва, дата початку рекламної кампанії, дата завершення рекламної кампанії, підсумковий масив рекламних показників, підсумковий масив показників рекламних каналів, підсумковий масив показників джерел для аналізу популярності рекламної продукції. Усі поля мають бути заповнені для збереження. За замовчуванням це робиться автоматично.

ActionLog – цей клас відповідає за збереження дій усіх користувачів. У ньому наявні такі поля: дата, логін, дія.

### *Контролери*

У мобільному застосунку наявні такі контролери:

- AuthorizationViewController;
- CreatorViewController;
- TabBarViewController;
- MainViewController;

- ProductViewController;
- ReportViewController;
- UserViewController;
- ActionViewController;
- AboutViewController;
- GraphViewController.

AuthorizationViewController – контролер, який відповідає за реєстрацію та авторизацію користувачів. Спочатку іде перевірка введених даних. Потім логін та пароль відправляються на сервер та, якщо користувача з таким логіном немає, то створюється новий користувач. В іншому випадку іде перевірка на збіг введеного користувачем пароля з паролем у БД.

CreatorViewController – контролер, який відповідає за створення файлу рекламної кампанії. Після отримання правильних даних контролер зберігає їх у БД.

TabBarViewController – контролер, який відповідає за відображення списку внизу екрана у вигляді чотирьох кнопок за допомогою яких можна пересуватися між сторінками «Особистий кабінет», «Головна сторінка», «Аналіз продукції» та «Звіт». За замовчуванням він відображає «Головна сторінка».

MainViewController – контролер, який відповідає за відображення процесу аналізу рекламної кампанії. Він підраховує всі необхідні показники.

ProductViewController – контролер, який відповідає за відображення процесу аналізу рекламної продукції.

ReportViewController – контролер, який відповідає за створення звіту по завершенні рекламної кампанії.

UserViewController – контролер, який відповідає за роботу особистого кабінету користувача.

ActionViewController – контролер, який відповідає за відображення журналу дій користувачів.

AboutViewController – контролер, який відповідає за відображення інформації про мобільний застосунок.

GraphViewController – контролер, який відповідає за детальний перегляд графіків. Після вибору необхідної інформації, контролер відобразить ці зміни на графіку.

### *Екрани*

У мобільному застосунку наявні такі екрани:

- AuthorizationView;
- CreatorView;
- TabBarView;
- MainView;
- ProductView;
- ReportView;
- UserView;
- ActionView;
- AboutView;
- GraphView.

AuthorizationView – на цьому екрані розташовані два поля для вводу даних від користувача «Логін» та «Пароль», які в якості даних приймають будь-які символи на клавіатурі, кнопка «Далі» та вікно з попереджувальними повідомленнями. Користувачу необхідно ввести дані у поля «Логін» та «Пароль» та натиснути кнопку «Далі». Якщо дані введені неправильно виведеться відповідне повідомлення інформуюче про це. В іншому випадку користувач перейде на наступний екран.

CreatorView – на цьому екрані розташовані три поля для вводу «Назва файлу», «Бюджет» і «Шлях до джерела» та один випадаючий список. Користувачу необхідно ввести дані у поля «Назва файлу», «Бюджет» та «Шлях до джерела» і вибрати щось одне з випадаючого списку та натиснути кнопку «Далі». Якщо дані введені неправильно виведеться відповідне

повідомлення інформуюче про це. В іншому випадку користувач перейде на наступний екран.

TabBarView – це єдиний екран, який не бачить користувач. Він необхідний лише на програмного рівні для реалізації переходу між сторінками «Особистий кабінет», «Головна сторінка», «Аналіз продукції» та «Звіт» за допомогою меню внизу.

MainView – на цьому екрані присутні графіки та текстові поля, за допомогою яких відображається процес аналізу рекламної кампанії. Якщо користувач натисне на графік він перейде на екран детального перегляду графіків.

ProductView – на цьому екрані присутній один випадаючий список, графік і текстові поля, за допомогою яких відображається процес аналізу рекламної продукції. Користувач має можливість вибрати необхідне джерело у випадаючому списку, а також натиснути на графік і перейти на екран детального перегляду графіків.

ReportView – на цьому екрані присутня кнопка «Завершити». Після того, як користувач її натисне процес аналізу рекламної кампанії буде завершено і виведеться вся необхідна інформація.

UserView – на цьому екрані розташовані п'ять полів для вводу даних від користувача: «логін», «пароль», «ім'я», «дата народження», «назва підприємства». Також тут присутня кнопка «Зберегти». Користувач має можливість змінювати інформацію про себе й після натискання кнопки «Зберегти» вона збережеться в БД. Якщо дані введені неправильно виведеться відповідне повідомлення інформуюче про це.

ActionView – екран, на якому відображається таблиця з трьома колонками «Дата», «Логін», «Дія». Тут відображаються дії всіх користувачів пов'язаних з одним файлом користувачам із роллю «Admin». Також, оскільки ця сторінка не входить до TabBarView, тут присутня кнопка «Назад» за допомогою якої можна повернутися на попередній екран.

AboutView – екран, на якому присутні текстові поля, які описують мобільний застосунок. Також, оскільки ця сторінка не входить до TabBarView, тут присутня кнопка «Назад» за допомогою якої можна повернутися на попередній екран.

GraphView – екран на якому присутній графік та випадючі списки для роботи з ним. Також, оскільки ця сторінка не входить до TabBarView, тут присутня кнопка «Назад» за допомогою якої можна повернутися на попередній екран.

### *Сервер*

Для зберігання та маніпулювання даними і доступу до них у БД був розроблений сервер на мові програмування Javascript. До нього входять усі необхідні функції та модулі для роботи з представленими класами.

Основні функції серверу:

- getUser;
- createUser;
- changeUser;
- createAdvertisementAnalyze;
- findAdvertisementAnalyze;
- createReport;
- findReport;
- addAction;
- findActionList.

GetUser – ця функція відповідає за пошук наявного у системі користувача. Вона приймає строку з логіном, після чого робить запит до БД. Якщо користувач знайдений повертає повідомлення з зашифрованим паролем. В іншому випадку повертає повідомлення «false». Усі вхідні дані є обов’язковими.



CreateUser – ця функція відповідає за створення нового користувача. Вона приймає логін та пароль у зашифрованому вигляді і, якщо користувача з таким логіном немає, записує їх у БД. Усі вхідні дані є обов’язковими.

ChangeUser – ця функція відповідає за зміну інформації в особистому кабінеті користувача. Вона приймає логін, пароль, ім’я, дата народження та назву підприємства, після чого виконується пошук у БД за наявним логіном і змінюються надані поля на значення, які було отримано. Усі вхідні дані не є обов’язковими та має бути заповненим хоча б одне поле.

CreateAdvertisementAnalyze – ця функція відповідає за створення файлу аналізу рекламної кампанії. Вона приймає назву файлу, бюджет та дату початку рекламної кампанії і, якщо файлу з такою назвою не існує, записує їх у БД. Усі вхідні дані є обов’язковими.

FindAdvertisementAnalyze – ця функція відповідає за пошук файлу аналізу рекламної кампанії. Вона приймає назву файлу, після чого виконує пошук у БД. Якщо заданий файл був знайдений повертає інформацію про нього. В іншому випадку повертає повідомлення «false». Усі вхідні дані є обов’язковими.

CreateReport – ця функція відповідає за створення звіту по завершенні рекламної кампанії. Вона приймає такі дані: назва, дата початку рекламної кампанії, дата завершення рекламної кампанії, підсумковий масив рекламних показників, підсумковий масив показників рекламних каналів, підсумковий масив показників джерел для аналізу популярності рекламної продукції. Після чого записує їх у БД. Усі вхідні дані є обов’язковими.

FindReport – ця функція відповідає за пошук звіту по рекламній кампанії. Вона приймає назву звіту і виконує пошук у БД. Якщо звіт був знайдений повертає його. В іншому випадку повертає повідомлення «false». Усі вхідні дані є обов’язковими.

AddAction – ця функція відповідає за створення дії користувача. Вона приймає логін, дату та дію і назву файлу і робить пошук у БД по назві. Якщо

файл знайдений записує дані у БД. В іншому випадку повертає повідомлення «false». Усі вхідні дані є обов’язковими.

FindActionList – ця функція відповідає за пошук листа з діями користувача. Вона приймає назву файлу і, якщо файл був знайдений повертає лист. В іншому випадку повертає повідомлення «false». Усі вхідні дані є обов’язковими.

### **3.5. Висновки до розділу**

У цьому розділі було проаналізовано загальний вигляд, архітектура та структуру програмного забезпечення для аналізу рекламної кампанії реалізованого у вигляді мобільного застосунку. Було визначено за допомогою чого і як саме працює мобільний застосунок. До того ж, були переглянуті і проаналізовані системні файли, БД і бібліотеки мобільного застосунку.

У тому числі, був переглянутий спосіб реалізації мобільного застосунку, а саме шаблон проєктування MVC, який на мові програмування Swift поділяє програму на три рівнозначні частини: класи, контролери та екрани.

Усі класи, контролери та екрани мобільного застосунку були детально переглянуті та проаналізовані. Було визначено за що відповідає кожний клас, контролер та екран і які функції він виконує.

До того ж, були проаналізовані основні функції серверу. Було визначено за що відповідає кожна функція і що саме вона виконує.

У результаті маємо повноцінний аналіз мобільного застосунку для аналізу рекламної кампанії.

## 4. АНАЛІЗ МОБІЛЬНОГО ЗАСТОСУНКУ

### 4.1. Тестування мобільного застосунку

Тестування мобільного застосунку є дуже важливим етапом під час та після його розроблення. Основним завданням тестування є порівняння того, як програма працює з тим, як вона має працювати. За допомогою тестування можна запобігти появі критичних помилок у мобільному застосунку [23].

У нашому випадку тестування виконувалося неодноразово та різними способами.

Під час розроблення мобільного застосунку після завершення роботи над функцією чи модулем вони тестувалися вручну розробником. За допомогою такого тестування кількість помилок на виході зменшилася втричі й помилки, які траплялися під час розроблення виправлялися миттєво.

До того ж, під час розроблення мобільного застосунку програміст дотримувався певних правил для полегшення тестування, а саме:

- модулі та функції мобільного застосунку мають працювати незалежно один від одного;
- можливість зміни переданих даних до модулів чи функцій, замість постійних значень;
- модулі та функції мобільного застосунку мають бути максимально прості та зрозумілі;
- назви змінних модулів та функцій мають бути максимально прості та зрозумілі;
- назви функцій та їхніх параметрів мають бути максимально прості та зрозумілі.

Після завершення розроблення мобільного застосунку були написані Unit тести. За допомогою них можна знайти помилки в тих чи інших модулях мобільного застосунку [24]. Проте, для їхньої роботи необхідно було надати доступ до повного функціонала мобільного

застосунку. Саме тому були створені Моск-об'єкти. Майже всі модулі мобільного застосунку пройшли цей різновид тестування, а ті, що не пройшли були виправлені.

Після успішного тестування коду мобільного застосунку було прийнято рішення надати доступ до мобільного застосунку невеликій групі людей для тестування інтерфейсу мобільного застосунку та виявленню критичних помилок під час роботи, тобто зробити так званий закритий бета-тест мобільного застосунку. Була створена група з десяти осіб, які були зацікавлені в мобільному застосунку. Їм був наданий доступ до мобільного застосунку на один день, після чого вони мали написати відгук, у якому потрібно було вказати проблеми в роботі з мобільним застосунком та побажання щодо зміни інтерфейсу.

Під час цього тестування було виявлено проблеми в роботі інтерфейсу на пристроях із малим розміром дисплею та критична помилка, із-за якої неможливо було користуватися мобільним застосунком. Вони були виправлені.

Отже, тестуючи мобільний застосунок багато разів, різними способами та на різних етапах вдалося позбутися 98% помилок у роботі з мобільним застосунком, у тому числі критичних, та покращити роботу інтерфейсу.

#### **4.2. Порівняння мобільного застосунку з наявними аналогами**

У першому розділі дипломного проєкту були проаналізовані наявні рішення поставленої проблеми і створена порівняльна таблиця з недоліками й перевагами наявних аналогів. Процес розроблення мобільного застосунку відбувався згідно вимог, описаних у підрозділі 1.4 та з урахуванням переваг та недоліків наявних аналогів.

У результаті було отримано мобільний застосунок, який :

- надає можливість користуватися їм одночасно декільком користувачам;

- має модулі та функції для ефективного аналізу рекламної кампанії;
- шифрує та зберігає всю необхідну інформацію в БД;
- надає можливість користуватися їм лише за умови проходження реєстрації за допомогою логіну та паролю.

Також, були виправлені недоліки наявних аналогів, описаних у підрозділі 1.2, а саме:

- цінова доступність;
- повнота функціональних можливостей;
- ефективність використання.

До того ж, оскільки програмне забезпечення реалізовано у вигляді мобільного застосунку воно є набагато зручнішим та більш перспективним у порівнянні з наявними аналогами.

У результаті маємо програмне забезпечення, що відповідає всім поставленим вимогам та є конкурентоспроможним.

#### **4.3. Пропозиції для майбутнього покращення мобільного застосунку**

Створений мобільний застосунок є мінімально працюючим і може мати більше можливостей у майбутньому. Внаслідок того, що потреби підприємств та рівень якості аналізу рекламної кампанії з кожним роком зростає постає питання у вдосконаленні створеного мобільного застосунку.

До того ж, під час закритого бета-тесту були сформовані рекомендації щодо покращення мобільного застосунку:

- розробити більший функціонал для користувачів із роллю «Admin»;
- зробити мінімальний функціонал для незареєстрованих користувачів;
- зробити більше способів реєстрації;
- додати меню для пошуку інформації в мобільному застосунку;
- зробити можливість коментування інформації користувачами;

- зробити можливість передачі інформації стороннім сервісам;
- додати push-повідомлення з інформуванням про ті чи інші події в мобільному застосунку;
- змінити цінову політику мобільного застосунку, на основі прибутку підприємства;
- додати чат для можливості спілкування користувачів, що мають спільний доступ до файлу між собою;
- додати декілька мов, доступних для роботи з мобільним застосунком;
- покращити графічну складову інтерфейсу.

У наступних версіях мобільного застосунку описані вище проблеми можуть бути виправлені.

#### **4.4. Висновки до розділу**

У цьому розділі було проаналізовано тестування мобільного застосунку для аналізу рекламної кампанії, його порівняння з аналогами та пропозиції для його майбутнього покращення.

Був розглянутий та проаналізований процес тестування мобільного застосунку. Тестування відбувалося різними способами, а саме: повне тестування, проміжне тестування, тестування користувачами. По завершенні розроблення певного модуля мобільного застосунку він тестувався вручну розробником. По завершенні розроблення мобільного застосунку були написанні Unit тести для його тестування. До того ж, був проведений закритий бета-тест мобільного застосунку. У результаті чого, кількість помилок у роботі з мобільним застосунком значно зменшилась.

Створений мобільний застосунок був порівняний з наявними аналогами. У результаті чого, було одержано програмне забезпечення, що реалізує поставлені вимоги та виправляє недоліки наявних аналогів.

До того ж, було звернено увагу на зауваження користувачів під час закритого тестування та сформований список вимог для покращення

мобільного застосунку в майбутньому. Серед критичних недоліків мобільного застосунку найважливішими для розроблення в майбутньому є більше способів реєстрації, функціонал для незареєстрованих користувачів та можливість передачі інформації стороннім сервісам.

## ВИСНОВКИ

У дипломному проєкті зроблене програмне забезпечення для аналізу рекламної кампанії у вигляді мобільного застосунку на операційну систему iOS. Мобільний застосунок розроблено для підприємств, які потребують якісного та дешевого аналізу рекламної кампанії, внаслідок чого їм можуть користуватися як великі підприємства так і малі, проєкти та стартапи.

У результаті дослідження обраної теми за допомогою перегляду основної інформації і проблематики щодо аналізу рекламної кампанії було прийнято рішення, що тема є актуальною на сьогодні.

Також, переглянувши наявні аналоги, що вирішують поставлену проблему, були знайдені їхні недоліки та переваги та створена порівняльна таблиця.

Після цього було прийнято рішення розробляти програмне забезпечення для аналізу рекламної кампанії у вигляді мобільного застосунку. До того ж, були сформовані вимоги до програмного забезпечення, серед яких є функціональні та нефункціональні вимоги, вимоги до якості, вимоги до безпеки.

Проаналізувавши можливі шляхи реалізації мобільних застосунків було визначено, що програмне забезпечення буде реалізовано на мобільну операційну систему iOS на мові програмування Swift з використанням БД MongoDB та сервером, написаним на мові програмування Javascript.

Розроблений мобільний застосунок має такі можливості:

- реєстрація нового користувача за допомогою логіну та паролю;
- авторизація наявного користувача;
- декілька рівнів доступів користувачів;
- створення файлу аналізу рекламної кампанії;
- перегляд аналізу рекламної кампанії;



- перегляд аналізу рекламної продукції;
- створення звіту по завершенні рекламної кампанії;
- особистий кабінет користувача;
- перегляд дій інших користувачів, що мають доступ до файлу;
- перегляд інформації про мобільний застосунок.

Мобільний застосунок розроблений та протестований згідно сформованих вимог. У результаті тестування виправлено 98% помилок, які траплялися при роботі з мобільним застосунком.

До того ж, був переглянутий подальший розвиток мобільного застосунку. Проаналізувавши ситуацію на ринку й зауваження користувачів під час закритого тестування були сформовані і проаналізовані вимоги для покращення програмного забезпечення в майбутньому.

У результаті ми маємо дешевий мобільний застосунок, який допомагає якісно аналізувати рекламну кампанію.

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Основы рекламы [Текст] / уклад. Ю.С. Бернадская, С.С. Марочкина, Л.Ф.Смотрова. ; за заг. ред. Л.М. Дмитриевой. — М. : ИВЦ “Наука”, 2005. — 281 с.
2. Рекламная деятельность: организация, планирование, оценка эффективности. [Текст] / уклад. Пономарёва А.М. — М. : ИВЦ “МарТ”, 2009. — 240с
3. Android (operating system): [Электронный ресурс]. — 2020. —  
Режим доступа до ресурсу:  
[https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). — (10.04.2020).
4. Google Play: [Электронный ресурс]. — 2020. — Режим доступа до  
ресурсу: [https://en.wikipedia.org/wiki/Android\\_Studio](https://en.wikipedia.org/wiki/Android_Studio). — (10.04.2020).
5. Android Studio: [Электронный ресурс]. — 2020. — Режим доступа до  
ресурсу: [https://en.wikipedia.org/wiki/Google\\_Play](https://en.wikipedia.org/wiki/Google_Play). — (10.04.2020).
6. App Store: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу:  
[https://en.wikipedia.org/wiki/App\\_Store\\_\(iOS\)](https://en.wikipedia.org/wiki/App_Store_(iOS)). — (10.04.2020).
7. Xcode 10: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу:  
<https://developer.apple.com/xcode/>. — (10.04.2020).
8. Swift (programming language): [Электронный ресурс]. — 2020. —  
Режим доступа до ресурсу:  
[https://en.wikipedia.org/wiki/Swift\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language)). —  
(10.04.2020).
9. Objective-C: [Электронный ресурс]. — 2020. — Режим доступа до  
ресурсу: <https://en.wikipedia.org/wiki/Objective-C>. — (10.04.2020).
10. Javascript: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу:  
<https://en.wikipedia.org/wiki/JavaScript>. — (12.04.2020).

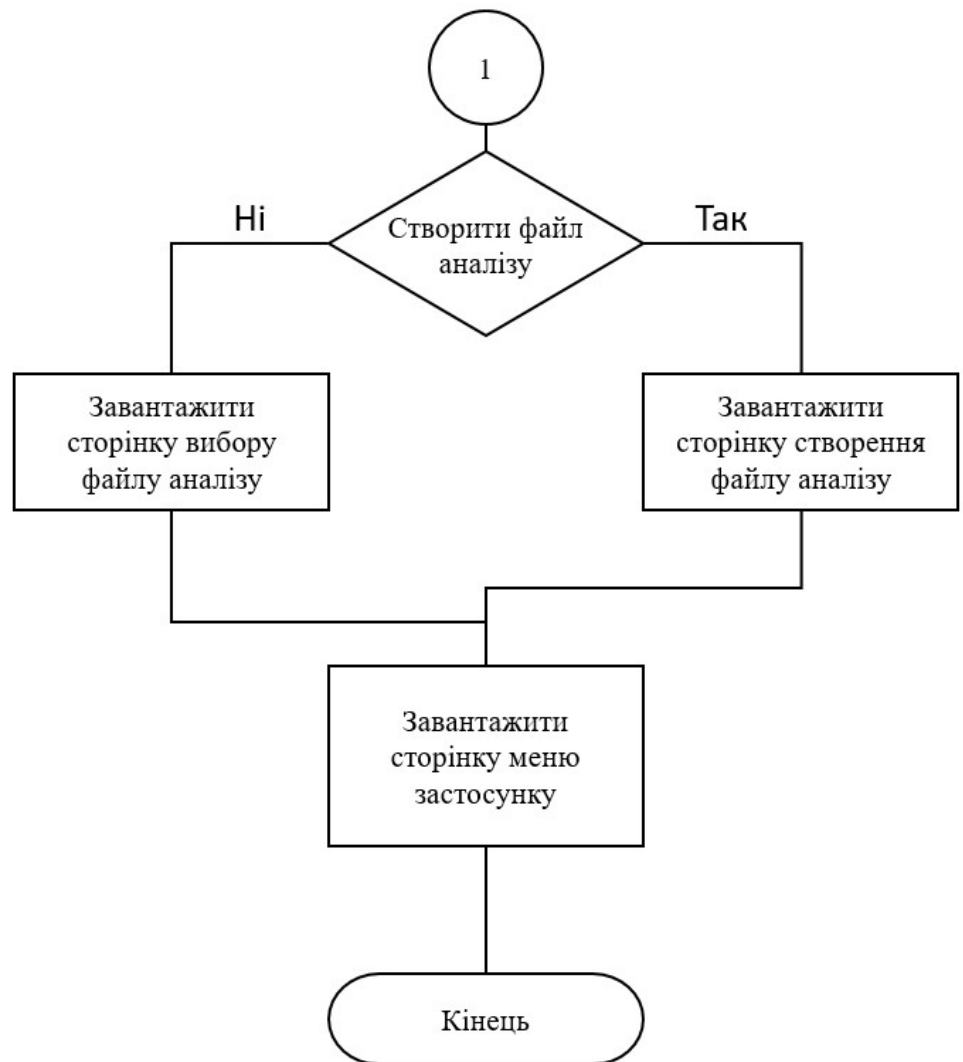
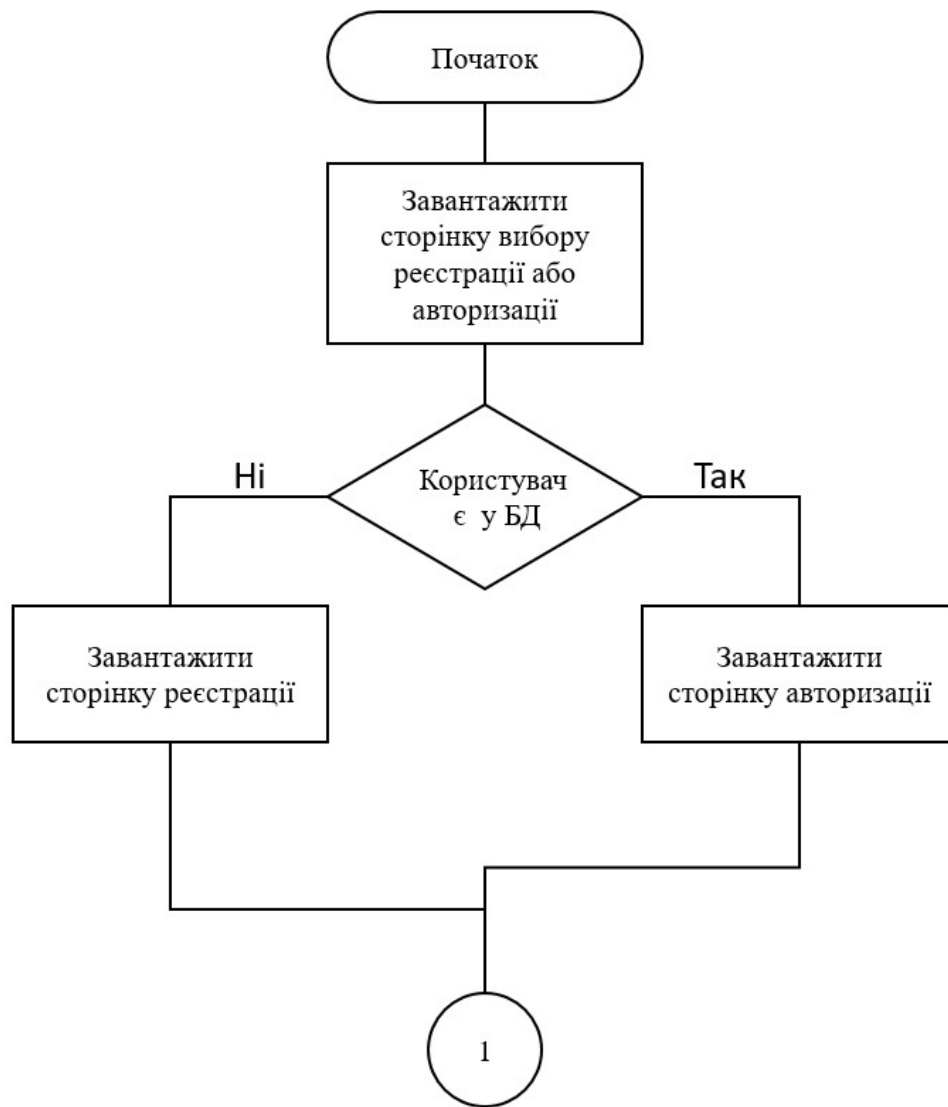
11. PHP: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://en.wikipedia.org/wiki/PHP>. — (12.04.2020).
12. Python: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/Python>. — (12.04.2020).
13. MySQL: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/MySQL>. — (12.04.2020).
14. MySQL: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Realm\\_\(database\)](https://en.wikipedia.org/wiki/Realm_(database)). — (12.04.2020).
15. MongoDB: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://en.wikipedia.org/wiki/MongoDB>. — (12.04.2020).
16. Model-view-controller: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://en.wikipedia.org/wiki/Model-view-controller>. — (07.05.2020).
17. Functional requirement: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Functional\\_requirement](https://en.wikipedia.org/wiki/Functional_requirement). — (07.05.2020).
18. Property list: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Property\\_list](https://en.wikipedia.org/wiki/Property_list). — (07.05.2020).
19. AppDelegate: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://developer.apple.com/documentation/uikit/uiapplicationdelegate>. — (07.05.2020).
20. Storyboard: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: <https://developer.apple.com/library/archive/documentation/General/Conceptual/Devpedia-CocoaApp/Storyboard.html>. — (07.05.2020).
21. Assets: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: [https://developer.apple.com/library/archive/documentation/Xcode/Reference/xcode\\_ref-Asset\\_Catalog\\_Format/index.html](https://developer.apple.com/library/archive/documentation/Xcode/Reference/xcode_ref-Asset_Catalog_Format/index.html). — (07.05.2020).

22. Launch Screen: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу:  
<https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/launch-screen/>..— (07.05.2020).
23. Software testing: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Software\\_testing](https://en.wikipedia.org/wiki/Software_testing). — (10.05.2020).
24. Unit testing: [Электронный ресурс]. — 2020. — Режим доступа до ресурсу: [https://en.wikipedia.org/wiki/Unit\\_testing](https://en.wikipedia.org/wiki/Unit_testing). — (10.05.2020).

## **ДОДАТКИ**

## **Додаток 1**

### **Копії графічних матеріалів**

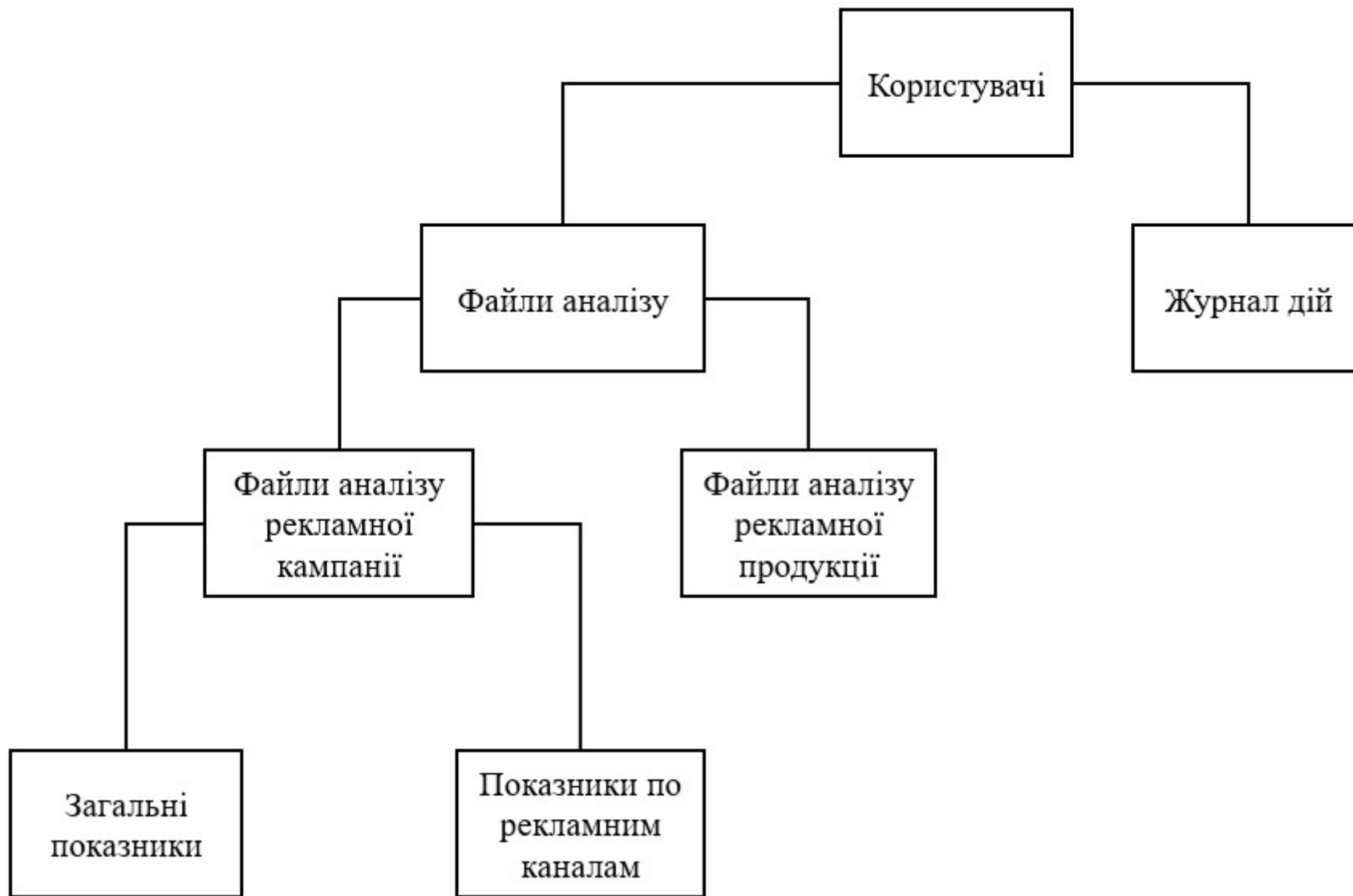


ДП.045440-06-99

Програмне забезпечення для аналізу рекламної кампанії

Алгоритм взаємодії графічних елементів.

Схема алгоритму



ДП.045440-07-99

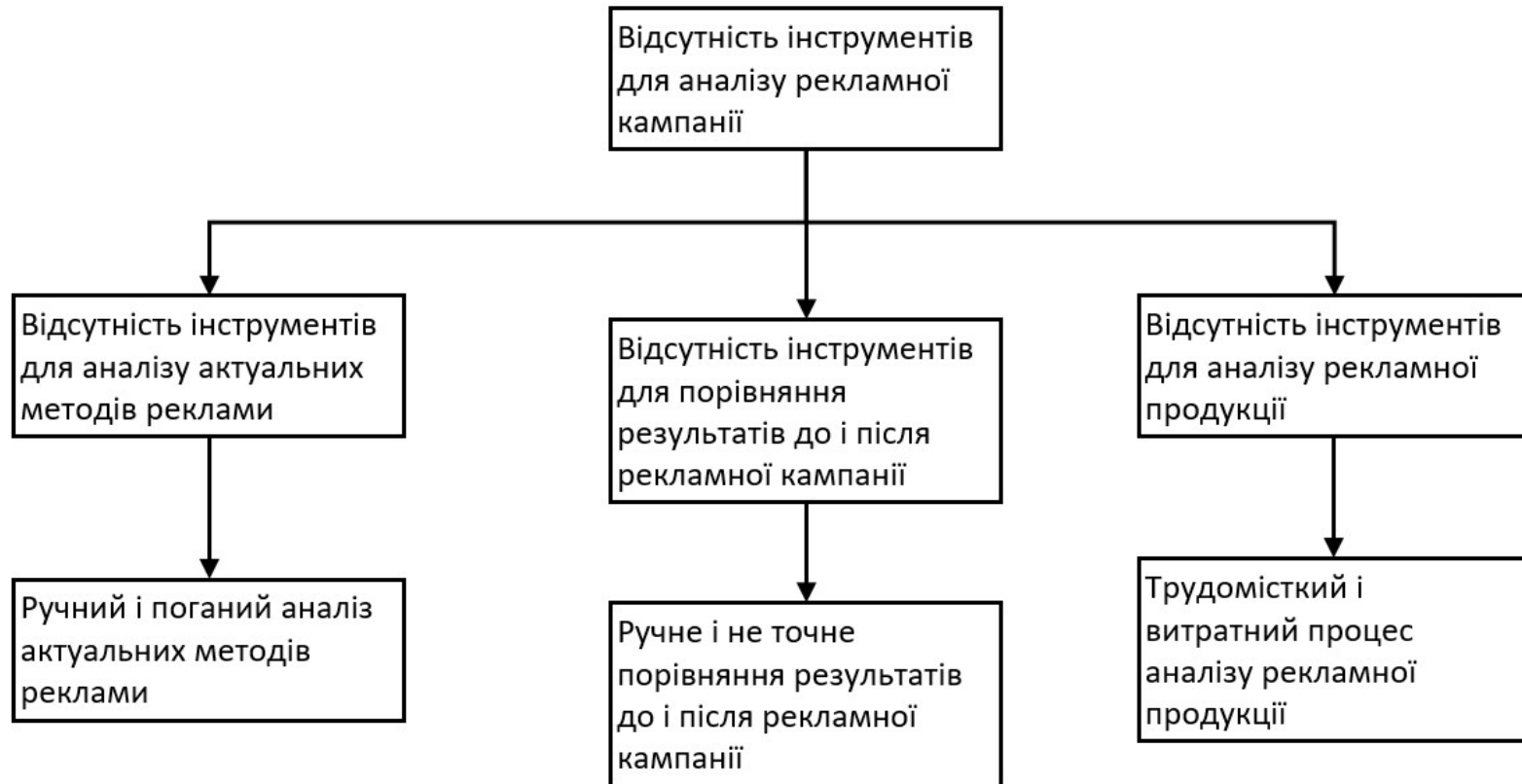
Програмне забезпечення для аналізу рекламної кампанії

База даних програмного забезпечення.

Схема даних

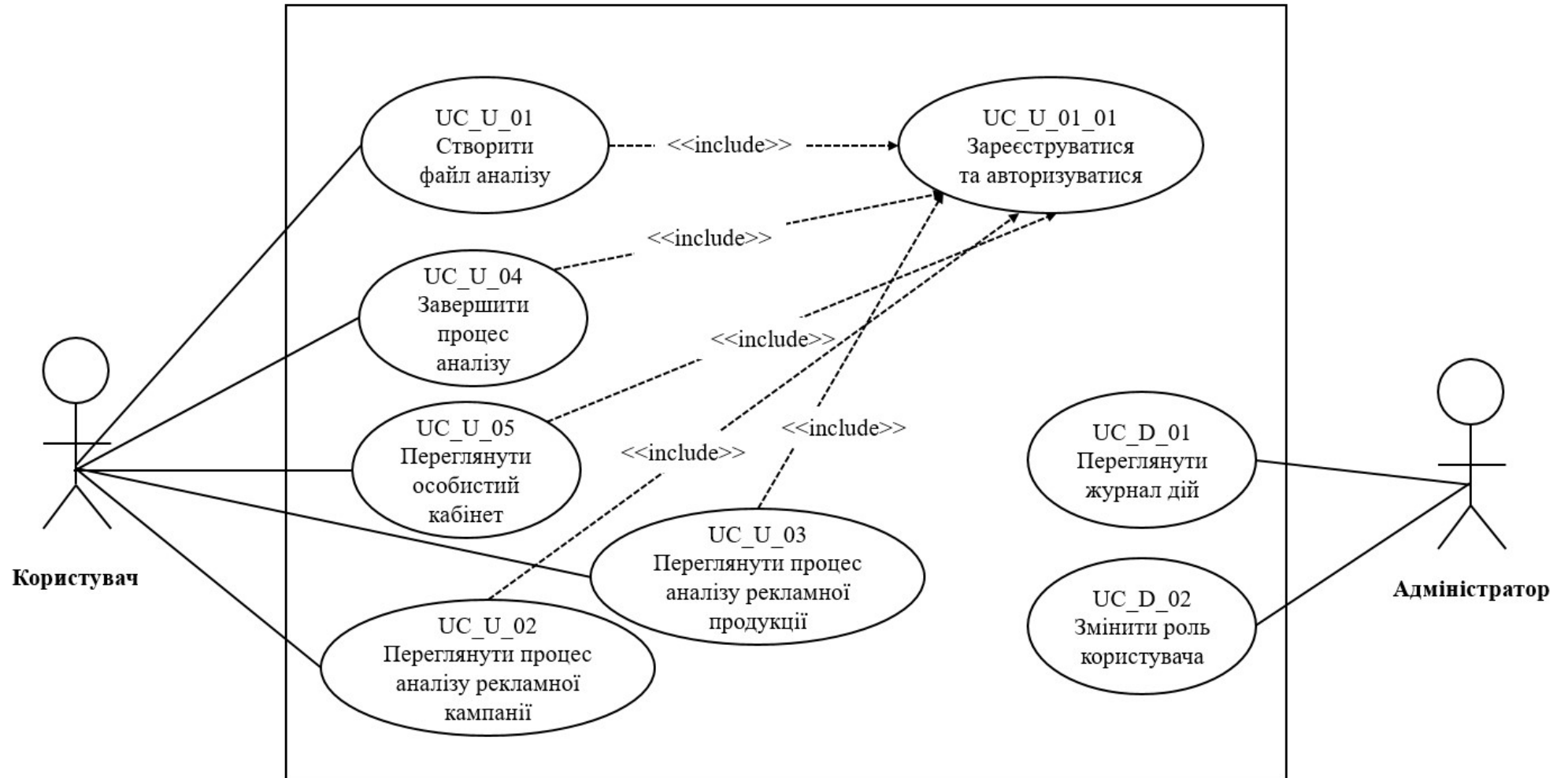


## Дерево проблем



Нечипоренко Владислав, група КП-61

## Діаграма діяльності програмного забезпечення



## **Додаток 2**

### **Лістинг програми**

## Основний контролер MainViewController.swift

```
import UIKit
import Charts
import SwiftDataTables

class MainViewController: UIViewController {

    @IBOutlet weak var startDate: UILabel!
    @IBOutlet weak var graph: LineChartView!
    @IBOutlet weak var graph2: LineChartView!
    @IBOutlet weak var graph3: LineChartView!
    @IBOutlet weak var graph4: LineChartView!
    var months = [String]()
    var unitsSold = [Double]()
    weak var axisFormatDelegate: IAxisValueFormatter?

    @IBAction func detail(_ sender: UIButton) {
        performSegue(withIdentifier: "bigChart", sender: self)
    }

    @IBAction func backFromModalMain(_ segue: UIStoryboardSegue) {
    }

    @IBOutlet weak var currentDate: UILabel!
    var Analyzer1 = Analyzer()

    func setCurrentDate(){
        let date = Date()
        let calendar = Calendar.current
        if Analyzer1.currentWeek == 1{
            currentDate.text = "1-
7.\(Analyzer1.month!).\(Analyzer1.year!)"
        }
        if Analyzer1.currentWeek == 2{
            currentDate.text = "8-
14.\(Analyzer1.month!).\(Analyzer1.year!)"
        }
        if Analyzer1.currentWeek == 3{
            currentDate.text = "15-
21.\(Analyzer1.month!).\(Analyzer1.year!)"
        }
        if Analyzer1.currentWeek == 4{
            currentDate.text = "22-
28.\(Analyzer1.month!).\(Analyzer1.year!)"
        }
        if Analyzer1.currentWeek == 5{
            currentDate.text = "29-\(calendar.component(.day, from:
date.endOfMonth)).\(\Analyzer1.month!).\(\Analyzer1.year!)"
        }
        startDate.text = "Дата початку:
\(\Analyzer1.startDate!).\(\Analyzer1.startMonth!).\(\Analyzer1.year!)"
    }
    var sem1 = DispatchSemaphore(value: 0)

    @IBOutlet weak var budgetView: UIStackView!

    func getCurrentDate(){
        let date = Date()
        let calendar = Calendar.current
        let result = calendar.component(.day, from: date)
```

```

        Analyzer1.month = calendar.component(.month, from: date)
        Analyzer1.year = calendar.component(.year, from: date)
        if result >= 1 && result <= 7{
            Analyzer1.currentWeek = 1
        }
        if result >= 8 && result <= 14{
            Analyzer1.currentWeek = 2
        }
        if result >= 15 && result <= 21{
            Analyzer1.currentWeek = 3
        }
        if result >= 22 && result <= 28{
            Analyzer1.currentWeek = 4
        }
        if result >= 29 && result <= 31{
            Analyzer1.currentWeek = 5
        }
    }

    @IBOutlet weak var clientsTotal: UILabel!

    @IBOutlet weak var clientsAverage: UILabel!

    @IBOutlet weak var salesTotal: UILabel!

    @IBOutlet weak var salesAverage: UILabel!

    @IBOutlet weak var profitTotal: UILabel!

    @IBOutlet weak var profitAverage: UILabel!

    @IBOutlet weak var budgetLabel: UILabel!

    @IBOutlet weak var roi: UILabel!

    @IBOutlet weak var ppc: UILabel!

    @IBOutlet weak var cpo: UILabel!

    @IBOutlet weak var cpl: UILabel!

    func calculateIndexes() {
        let roiResult: Double = Double(Analyzer1.calculateProfit() -
Analyzer1.calculateMoneyWaste())/Double(Analyzer1.calculateMoneyWaste())*10
0
        let ppcResult: Double =
(Double(Analyzer1.calculateMoneyWaste())/Double(Analyzer1.calculateNewClie
nts()))
        let cpoResult: Double =
(Double(Analyzer1.calculateMoneyWaste())/Double(Analyzer1.calculateSales()
))
        let cplResult: Double =
(Double(Analyzer1.calculateProfit())/Double(Analyzer1.calculateNewClients(
)))
        let ppcRounded = Double(round(100*ppcResult)/100)
        let cpoRounded = Double(round(100*cpoResult)/100)
        let cplRounded = Double(round(100*cplResult)/100)
        roi.text = "ROI: \(Int(roiResult)) %"
        ppc.text = "PPC: \(ppcRounded) $"
        cpo.text = "CPO: \(cpoRounded) $"
        cpl.text = "CPL: \(cplRounded) $"
    }
    func calculateLabels(){
        var clients = 0

```

```

var sales = 0
var profit = 0
var moneyWaste = 0
for x in Analyzer1.arrayToShow{
    clients += x.newClients
    sales += x.sales
    profit += x.profit
}
for x in Analyzer1.totalIndexes{
    if x.year == Analyzer1.startYear {
        if x.month == Analyzer1.startMonth{
            if x.week >= Analyzer1.startWeek{
                moneyWaste += x.moneyWaste
            }
        }
        if x.month > Analyzer1.startMonth
        {
            moneyWaste += x.moneyWaste
        }
    }
    if x.year > Analyzer1.startYear {
        moneyWaste += x.moneyWaste
    }
}
clientsTotal.text = "Всього за період: \$(clients)"
clientsAverage.text = "В середньому за день:
\$(clients/Analyzer1.arrayToShow.count)"
salesTotal.text = "Всього за період: \$(sales)"
salesAverage.text = "В середньому за день:
\$(sales/Analyzer1.arrayToShow.count)"
profitTotal.text = "Всього за період: \$(profit)"
profitAverage.text = "В середньому за день:
\$(profit/Analyzer1.arrayToShow.count)"
budgetLabel.text = "Бюджет: \$(Analyzer1.budget - moneyWaste)$"
}
var User1 = User()
override func viewDidLoad() {
    super.viewDidLoad()
    //addData()
    let data = ["name" : Analyzer1.sourceName!, "dbname" :
Analyzer1.bdName!, "collection" : Analyzer1.tableName!]
    server8(data)
    sem1.wait()
    addLabelsToBudgetView()
    getCurrentDate()
    makeDays()
    setCurrentDate()
    Analyzer1.calculateTotal()
    Analyzer1.makeArrayToShow()
    calculateLabels()
    Analyzer1.makeClientArray()
    Analyzer1.makeSalesArray()
    Analyzer1.makeProfitArray()
    Analyzer1.makeMoneyWasteArray()
    Analyzer1.makeSourcesArray(Analyzer1.currentWeek,
Analyzer1.year, Analyzer1.month)
    setMultipleChart()
    calculateIndexes()
    //    Analyzer1.startMonth = 5
    //    Analyzer1.startYear = 2020
    //    Analyzer1.startWeek = 4
    setChart(dataEntryY: Analyzer1.clientsArray, "Нових клієнтів", 1)
    setChart(dataEntryY: Analyzer1.salesArray, "Проданих товарів", 2)
    setChart(dataEntryY: Analyzer1.profitArray, "Прибуток", 3)
}

```

```

    }

    override func viewWillAppear(_ animated: Bool) {
        var Analyzer2 = Analyzer()
        if let tbc = tabBarController as? TabBarViewController {
            Analyzer2 = tbc.Analyzer1
            User1 = tbc.User1
        }
        Analyzer1.actionList = Analyzer2.actionList
        Analyzer1.addAction(User1.login, Date(), "Перейшов на сторінку
аналізу рекламної кампанії")
        updateList()
    }

    func updateList() {
        var actions = [String]()
        for x in Analyzer1.actionList{
            actions.append(x.action)
        }
        let data2 = ["name" : Analyzer1.name! , "actions" : actions] as
[String : Any]
        server6(data2)
    }

    override func viewWillDisappear(_ animated: Bool) {
        if let tbc = tabBarController as? TabBarViewController {
            tbc.Analyzer1 = Analyzer1
        }
    }

    var days = [String]()

    func getCountOfDays() -> Int{
        let dateComponents = DateComponents(year: Analyzer1.year, month:
Analyzer1.month)
        let calendar = Calendar.current
        let date = calendar.date(from: dateComponents)!

        let range = calendar.range(of: .day, in: .month, for: date)!
        let numDays = range.count
        return numDays
    }

    func addLabelsToBudgetView() {
        Analyzer1.sourceIndexes.sort{
            $0.source < $1.source
        }
        var a = Analyzer1.sourceIndexes[0].source
        var b = Analyzer1.sourceIndexes[0]
        var moneyWaste = 0
        for (id,x) in Analyzer1.sourceIndexes.enumerated(){
            if x.source != a || id == Analyzer1.sourceIndexes.endIndex-
1{
                if id == Analyzer1.sourceIndexes.endIndex-1 {
                    moneyWaste += x.moneyWaste
                }
                let label = UILabel()
                label.textColor = UIColor.black
                label.text = "\(a!): \(b.budget - moneyWaste)$"
                label.font = UIFont(name: "Times New Roman", size: 20)
                budgetView.addArrangedSubview(label)
                moneyWaste = 0
            }
        }
    }

```

```

        if x.year == Analyzer1.startYear {
            if x.month == Analyzer1.startMonth{
                if x.week >= Analyzer1.startWeek{
                    moneyWaste += x.moneyWaste
                }
            }
            if x.month > Analyzer1.startMonth
            {
                moneyWaste += x.moneyWaste
            }
        }
        if x.year > Analyzer1.startYear {
            moneyWaste += x.moneyWaste
        }
        a = x.source
        b = x
    }
    budgetView.sizeToFit()
    budgetView.layoutIfNeeded()
}

func makeDays() {
    var startDate = 1
    for x in 1...getCountOfDays() {
        days.append(String(x))
    }
    if Analyzer1.currentWeek == 1{
        startDate = 1
    }
    if Analyzer1.currentWeek == 2{
        startDate = 8
    }
    if Analyzer1.currentWeek == 3{
        startDate = 15
    }
    if Analyzer1.currentWeek == 4{
        startDate = 22
    }
    if Analyzer1.currentWeek == 5{
        startDate = 29
    }
    for x in startDate-1...days.count-1{
        months.append(days[x])
    }
    // print(months)
}

func setChart(dataEntryY forY: [Int], _ label: String, _ currentGraph:
Int) {
    axisFormatDelegate = self
    self.navigationController?.setNavigationBarHidden(true,
animated: true)
    var dataEntries:[ChartDataEntry] = []
    for i in 0..

```



```

var xAxisValue = graph.xAxis
var yAxisValue = graph.leftAxis
let fmt = NumberFormatter()
fmt.numberStyle = .decimal
fmt.maximumFractionDigits = 0
fmt.groupingSeparator = ","
fmt.decimalSeparator = "."
if currentGraph == 1{
    graph.data = chartData
    graph.translatesAutoresizingMaskIntoConstraints = false
    graph.noDataText = "Немає даних"
    xAxisValue = graph.xAxis
    yAxisValue = graph.leftAxis
}
if currentGraph == 2{
    graph2.data = chartData
    graph2.translatesAutoresizingMaskIntoConstraints = false
    graph2.noDataText = "Немає даних"
    xAxisValue = graph2.xAxis
    yAxisValue = graph2.leftAxis
}
if currentGraph == 3{
    graph3.data = chartData
    graph3.translatesAutoresizingMaskIntoConstraints = false
    graph3.noDataText = "Немає даних"
    xAxisValue = graph3.xAxis
    yAxisValue = graph3.leftAxis
}
yAxisValue.valueFormatter =
DefaultAxisValueFormatter.init(formatter: fmt)
xAxisValue.valueFormatter = axisFormatDelegate
xAxisValue.labelCount = forY.count-1

}

func setDataEntry(_ values: [String:[Int]]) -> LineChartData{
    var circleColors: [UIColor] = []
    axisFormatDelegate = self
    self.navigationController?.setNavigationBarHidden(true,
animated: true)
    var dataEntries:[ChartDataEntry] = []
    let data = LineChartData()
    for x in values{
        for i in 0..

```

```

        dataEntries = []
        circleColors = []
    }
    return data
}

func setMultipleChart() {
    graph4.noDataText = "Немає даних"
    var values = [String:[Int]]()
    for x in Analyzer1.sourcesData{
        values[x.key] = x.value["profit"]
    }
    var data = LineChartData()
    data = setDataEntry(values)
    graph4.data = data
    let xAxisValue = graph4.xAxis
    let yAxisValue = graph4.leftAxis
    let fmt = NumberFormatter()
    fmt.numberStyle = .decimal
    fmt.maximumFractionDigits = 0
    fmt.groupingSeparator = ","
    fmt.decimalSeparator = "."
    yAxisValue.valueFormatter =
DefaultAxisValueFormatter.init(formatter: fmt)
    xAxisValue.valueFormatter = axisFormatDelegate
    xAxisValue.labelCount = values["Google2"]!.count-1
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if let BigChartVC = segue.destination as?
BigChartViewController{
        BigChartVC.Analyzer1 = Analyzer1
    }
}

func server6 (_ dataDictionary: [String : Any]) {
    let url = URL(string: "http://localhost:3000/updateList")!
    var request = URLRequest(url: url)
    // request.setValue("application/x-www-form-urlencoded",
forHTTPHeaderField: "Content-Type")
    request.setValue("application/json", forHTTPHeaderField:
"Content-Type")
    request.setValue("application/json", forHTTPHeaderField:
"Accept")
    request.httpMethod = "POST"
    request.httpBody = try! JSONSerialization.data(withJSONObject:
dataDictionary)

    let task = URLSession.shared.dataTask(with: request) { data,
response, error in
        guard let _ = data,
            let response = response as? HTTPURLResponse,
            error == nil else {
// check for fundamental networking error
            print("error", error ?? "Unknown error")
            return
        }

        guard (200 ... 299) ~= response.statusCode else {
// check for http errors
            print("statusCode should be 2xx, but is
\(response.statusCode)")

```

```

        print("response = \(response)")
        return
    }
    //let responseString = String(data: data, encoding: .utf8)

    //let responseString = String(data: data, encoding: .utf8)
    //print("responseString = \(String(describing:
responseString))")
    }
    task.resume()
}

func server7 (_ dataDictionary: [String : Any]) {
    let url = URL(string: "http://localhost:3000/addData")!
    var request = URLRequest(url: url)
    // request.setValue("application/x-www-form-urlencoded",
forHTTPHeaderField: "Content-Type")
    request.setValue("application/json", forHTTPHeaderField:
"Content-Type")
    request.setValue("application/json", forHTTPHeaderField:
"Accept")
    request.httpMethod = "POST"
    request.httpBody = try! JSONSerialization.data(withJSONObject:
dataDictionary)

    let task = URLSession.shared.dataTask(with: request) { data,
response, error in
        guard let _ = data,
            let response = response as? HTTPURLResponse,
            error == nil else {
            // check for fundamental networking error
            print("error", error ?? "Unknown error")
            return
        }

        guard (200 ... 299) ~= response.statusCode else {
            // check for http errors
            print("statusCode should be 2xx, but is
\(response.statusCode)")
            print("response = \(response)")
            return
        }
        //let responseString = String(data: data, encoding: .utf8)
        self.sem1.signal()
        //let responseString = String(data: data, encoding: .utf8)
        //print("responseString = \(String(describing:
responseString))")
    }
    task.resume()
}

func server8 (_ dataDictionary: [String : Any]) {
    let url = URL(string: "http://localhost:3000/findData")!
    var request = URLRequest(url: url)
    // request.setValue("application/x-www-form-urlencoded",
forHTTPHeaderField: "Content-Type")
    request.setValue("application/json", forHTTPHeaderField:
"Content-Type")
    request.setValue("application/json", forHTTPHeaderField:
"Accept")
    request.httpMethod = "POST"

```

```

        request.httpBody = try! JSONSerialization.data(withJSONObject:
dataDictionary)

        let task = URLSession.shared.dataTask(with: request) { data,
response, error in
            guard let data = data,
                let response = response as? HTTPURLResponse,
                error == nil else {
// check for fundamental networking error
                print("error", error ?? "Unknown error")
                return
            }

            guard (200 ... 299) ~= response.statusCode else {
// check for http errors
                print("statusCode should be 2xx, but is
\(response.statusCode)")
                print("response = \(response)")
                return
            }
            //let responseString = String(data: data, encoding: .utf8)
            do {
                let json = try JSONSerialization.jsonObject(with: data,
options: [])
                if let array = json as? [[String : Any]] {
                    // print(json)
                    for k in array {
                        let month = k["month"] as? Int
                        let date = k["date"] as? Int
                        let newClients =
k[self.Analyzer1.newClientsInDB] as? Int
                        let moneyWaste = k[self.Analyzer1.wasteInDB] as?
Int
                        let name = k[self.Analyzer1.sourceNameInDB] as?
String
                        let profit = k[self.Analyzer1.profitInDB] as?
Int
                        let year = k["year"] as? Int
                        let sales = k[self.Analyzer1.salesInDB] as? Int
                        let week = k["week"] as? Int
                        self.Analyzer1.addToSourceIndexes(week!,
self.Analyzer1.budget/3, date!, year!, month!, name!, newClients!, sales!,
profit!, moneyWaste!)
                    }
                }
            } catch {
                print("Couldn't parse json \(error)")
            }
            self.sem1.signal()
            //let responseString = String(data: data, encoding: .utf8)
            //print("responseString = \(String(describing:
responseString))")
        }
        task.resume()
    }

}

extension MainViewController: IAxisValueFormatter {

```

```

        func stringValue(_ value: Double, axis: AxisBase?) -> String {
            return months[Int(value)]
        }
    }

    extension Date {
        var startOfDay: Date {
            return Calendar.current.startOfDay(for: self)
        }

        var startOfMonth: Date {

            let calendar = Calendar(identifier: .gregorian)
            let components = calendar.dateComponents([.year, .month], from:
self)

            return calendar.date(from: components)!
        }

        var endOfDay: Date {
            var components = DateComponents()
            components.day = 1
            components.second = -1
            return Calendar.current.date(byAdding: components, to:
startOfDay)!
        }

        var endOfMonth: Date {
            var components = DateComponents()
            components.month = 1
            components.second = -1
            return Calendar(identifier: .gregorian).date(byAdding:
components, to: startOfMonth)!
        }

        func isMonday() -> Bool {
            let calendar = Calendar(identifier: .gregorian)
            let components = calendar.dateComponents([.weekday], from: self)
            return components.weekday == 2
        }
    }
}

```

**Додаток 3**  
**Копія презентації**

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ  
КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ  
ДЛЯ АНАЛІЗУ РЕКЛАМНОЇ КАМПАНІЇ

Виконав: Нечипоренко Владислав Олександрович

Керівник: старший викладач кафедри ПЗКС, к.т.н., Люшенко Л.А.

Київ – 2020



## ПОСТАНОВКА ЗАДАЧІ

**Мета проєкту:** розробити програмне забезпечення для аналізу рекламної кампанії.

**Завдання:**

1. Проаналізувати предметну область, визначити її актуальність та проблематику, а також наявні програмні рішення та визначити вимоги до розроблення програмного забезпечення.
2. Розробити програмне забезпечення згідно вимог за допомогою обраних засобів реалізації.
3. Протестувати розроблене програмне забезпечення та проаналізувати його відповідність до вимог.





## АКТУАЛЬНІСТЬ

У наш час багато підприємств вкладають великі кошти в розвиток своєї продукції й досить значні грошові ресурси вкладаються в рекламну кампанію. Наявні рішення не вирішують поставлене завдання і розраховані для великих підприємств. Програмне забезпечення повинно бути доступним, тому мобільний застосунок – найбільш оптимальне рішення.

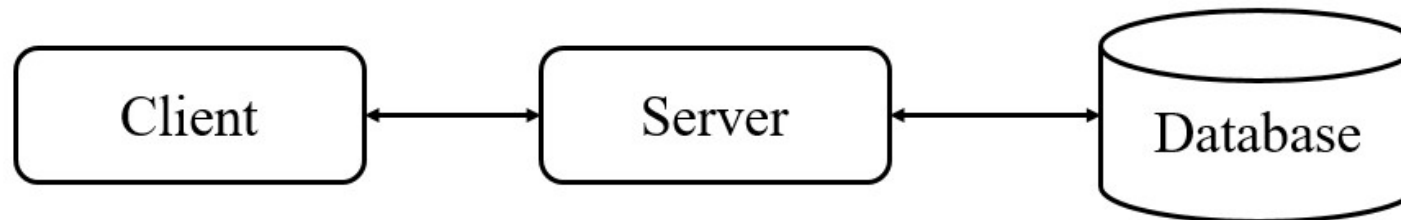
Тож, розробка програмного забезпечення для аналізу рекламної кампанії є актуальним рішенням сьогодні.

# НЕДОЛІКИ І ПЕРЕВАГИ АНАЛОГІВ



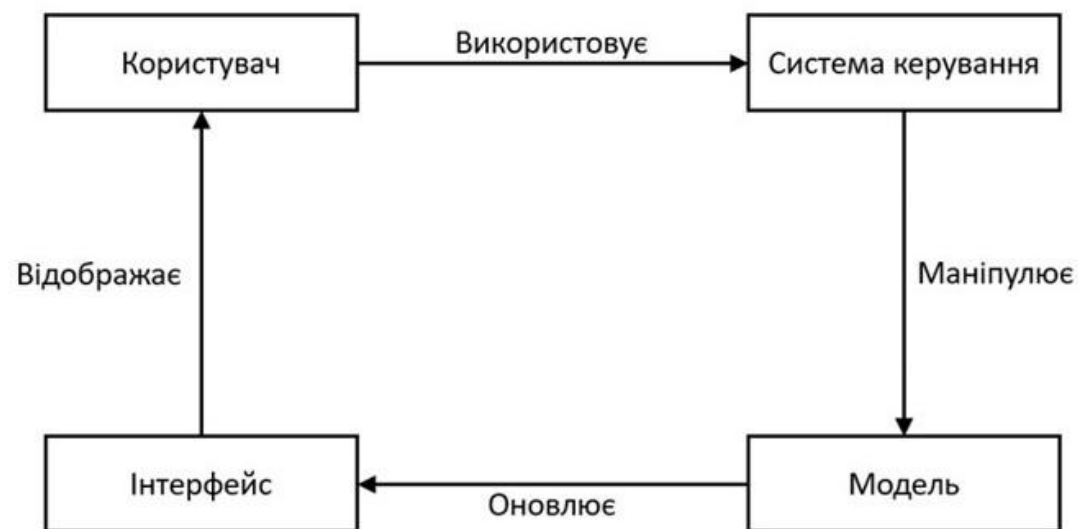
Показник	Google AdWords	Яндекс.Дірект	Roistat
Цінова доступність	✓	✓	✗
Швидкість	✓	✓	✗
Повнота функціональних можливостей	✗	✗	✓
Зручність	✓	✓	✗
Доступність	✓	✗	✓
Простота користування	✓	✓	✗
Гнучкість налаштування	✓	✓	✓
Ефективність використання	✗	✗	✓

# АРХІТЕКТУРА СИСТЕМИ



# ШАБЛОН ПРОЄКТУВАННЯ

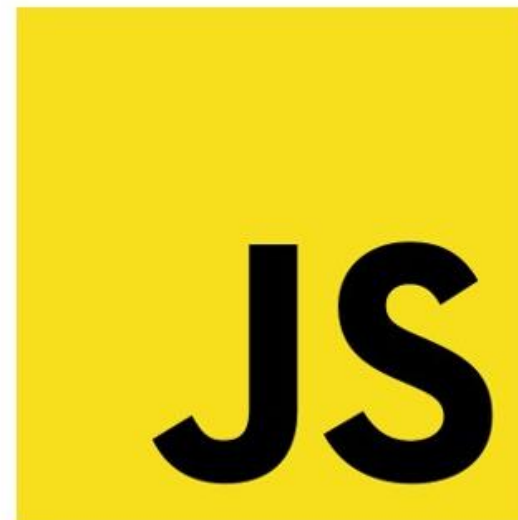
Програмне забезпечення реалізовано за допомогою шаблону проєктування MVC.



## ЗАСОБИ РОЗРОБЛЕННЯ

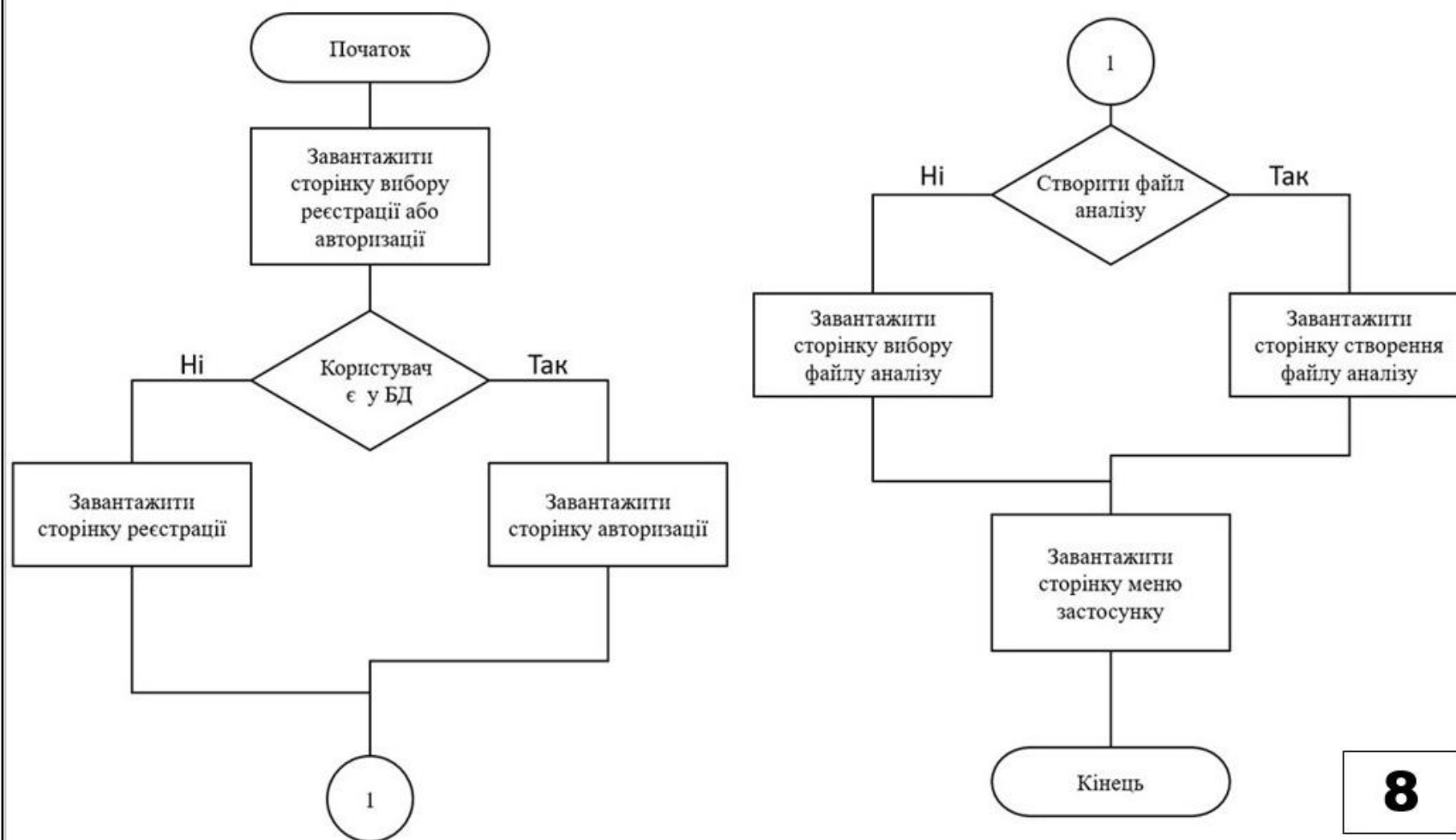


Swift

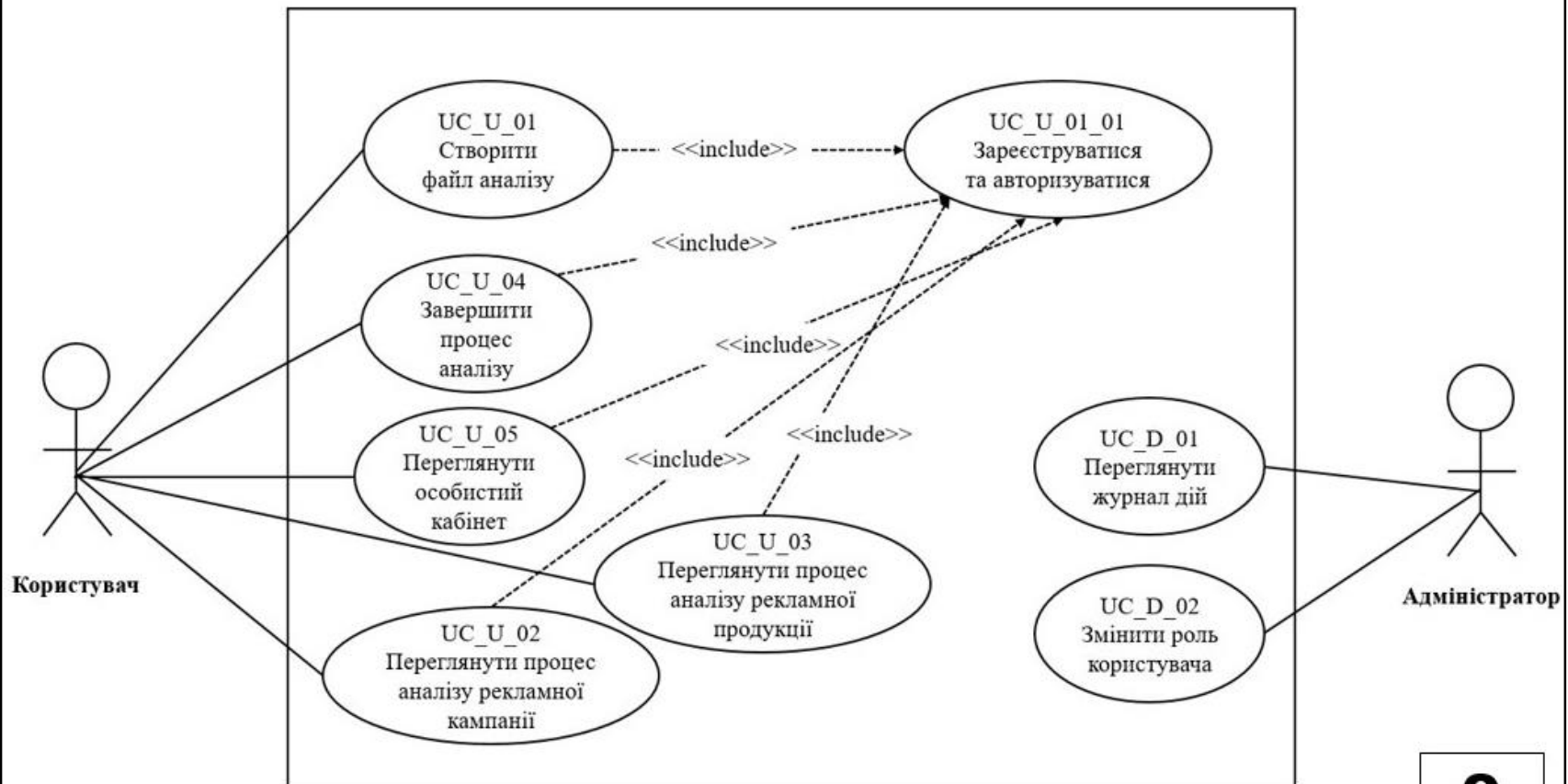


mongoDB

# АЛГОРИТМ ВЗАЄМОДІЇ ГРАФІЧНИХ ЕЛЕМЕНТІВ



# ДІАГРАМА ДІЯЛЬНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## ВІДПОВІДНІСТЬ РОЗРОБЛЕНИМ ВИМОГАМ ДО ПЗ



- реєстрація чи авторизація в системі за допомогою логіну та паролю;
- створення чи завантаження файлу аналізу рекламної кампанії;
- перегляд процесу аналізу рекламної кампанії;
- перегляд процесу аналізу рекламної продукції;
- детальний перегляд графіків з додатковими налаштуваннями;
- редагування даних в особистому кабінеті;
- сервер для збору необхідної інформації;





## ВІДПОВІДНІСТЬ РОЗРОБЛЕНИМ ВИМОГАМ ДО ПЗ

- шифрування важливої інформації;
- зручний та адаптивний інтерфейс;
- декілька рівнів доступу користувачів;
- аналітика рекламних каналів;
- журнал дій для користувачів з роллю «Admin»;
- функціонування одночасно з більше ніж двома користувачами;
- збереження важливої інформації у базі даних;
- можливість прочитати інформацію про розроблену систему.

# ІНТЕРФЕЙС ПЗ



12:41

Реєстрація Авторизація

Логін

Пароль

Ім'я

Дата нар.

Компанія

Далі

12:42

Реєстрація Авторизація

Логін

Пароль

Далі

# ІНТЕРФЕЙС ПЗ



12:42

Створити Завантажити

Назва

Бюджет

Джерело

Шлях

Назви полів:

Дата

Назва джерела

Нових клієнтів

Проданих товарів

Прибуток

Витрати

Назва продукту

Назва БД

Назва таблиці

Далі

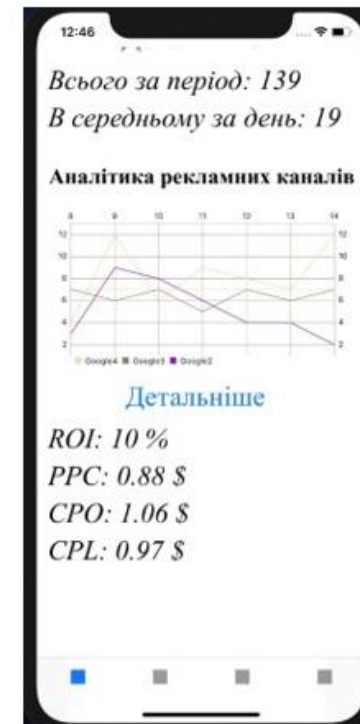
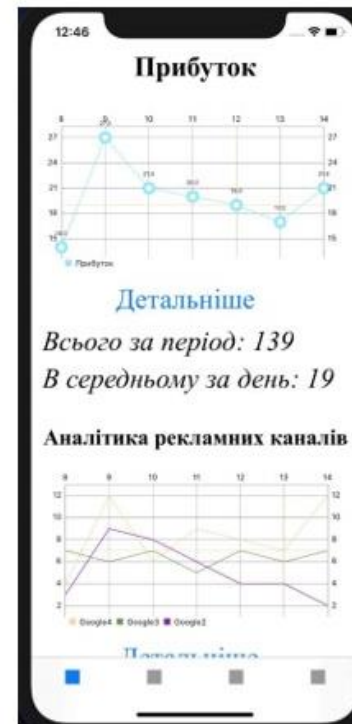
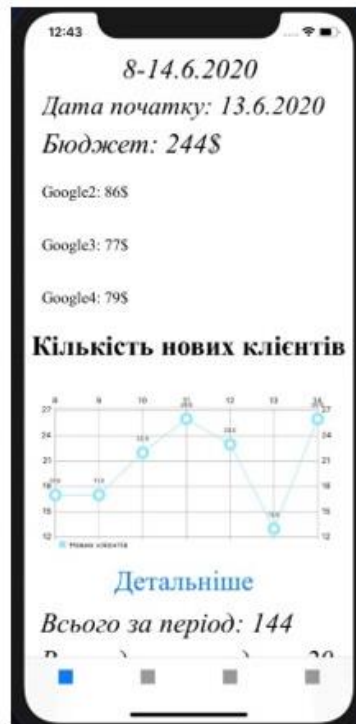
12:42

Створити Завантажити

Назва

Далі

# ІНТЕРФЕЙС ПЗ



# ІНТЕРФЕЙС ПЗ



12:54

Найкращий рекламний канал

Google2

Найгірший рекламний канал

Google3

ROI: +8 %  
PPC: -0.13 \$  
CPO: -0.37 \$  
CPL: -0.07 \$

Завершити

12:47

### Особистий кабінет

Логін: VladVlad

Пароль: \*\*\*\*\*

Ім'я: Daf

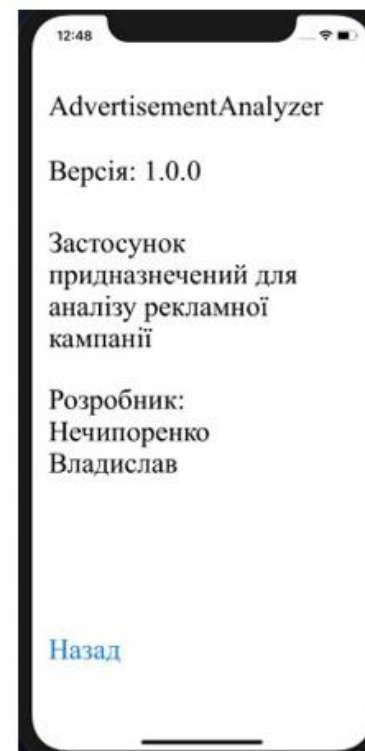
Дата нар.: 13.03.99

Компанія: Daf

Журнал дій

Інфо Оновити

# ІНТЕРФЕЙС ПЗ



# ІНТЕРФЕЙС ПЗ

12:48

Рік

Місяць

Тиждень

Показник

Джерело

ROI:

PPC:

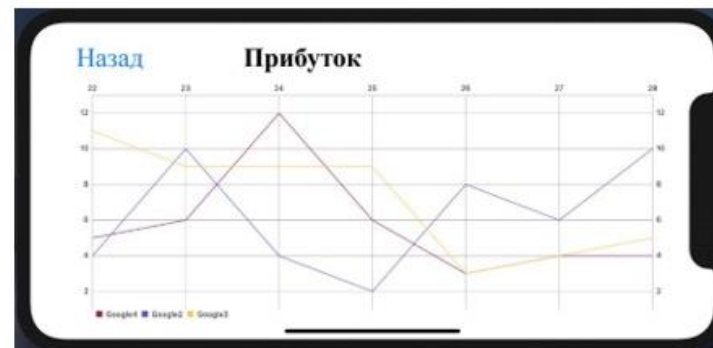
CPO:

CPL:

Назад

Оновити

Далі



12:58

Рік

Місяць

Тиждень

Показник

Джерело

ROI: 25 %

PPC: 0.76 \$

CPO: 1.84 \$

CPL: 0.96 \$

Назад

Оновити

Далі

# РЕЗУЛЬТАТИ ТЕСТУВАННЯ



1:33

Регістрація Авторизація

Логін

Пароль

Ім'я

Дата нар.

Компанія

Далі

12:47 PM

Регістрація Ав...ція

Логін

Пароль

Ім'я

Дата...

Комп...

Далі



1:33

Регістрація Авторизація

Логін

Пароль

Ім'я

Дата нар.

Компанія

Далі

1:31 PM

Регістрація Авторизація

Логін

Пароль

Ім'я

Дата нар.

Компанія

Далі







## РЕКОМЕНДАЦІЇ ДЛЯ ПОКРАЩЕННЯ

- розробити більший функціонал для користувачів із роллю «Admin»;
- зробити мінімальний функціонал для незареєстрованих користувачів;
- зробити більше способів реєстрації;
- додати меню для пошуку інформації в мобільному застосунку;
- зробити можливість передачі інформації стороннім сервісам;
- додати push-повідомлення з інформуванням про ті чи інші події в мобільному застосунку;
- додати чат для можливості спілкування користувачів, що мають спільний доступ до файлу між собою;
- додати декілька мов, доступних для роботи з мобільним застосунком.



## ВИСНОВКИ

1. Був проведений аналіз існуючих програмних рішень.
2. Запропоновано архітектуру ПЗ на основі шаблону проектування MVC.
3. Були визначені функціональні вимоги до розроблюваної системи, серед яких є апаратні вимоги, вимоги до інтерфейсу, вимоги до технологій.
4. Розроблено ПЗ, яке задовольняє критерії якості ПЗ.
5. Протестовано та порівняно ПЗ з аналогами за критеріями, які були визначені при аналізі існуючих рішень (переваги та недоліки).
6. Мобільний застосунок розроблений в повному обсязі з відповідністю до розроблених вимог. Тестування мобільного застосунку виконано в повному обсязі згідно наявної методики тестування.
7. Розроблений мобільний застосунок допомагає підприємствам чи звичайним користувачам економити свій час та аналізувати рекламну кампанію у своєму смартфоні.

# АНТИПЛАГІАТ



Власник документу:  
Люшенко Леся Анатоліївна gmail

ID перевірки:  
1004010593

Дата перевірки:  
12.06.2020 21:19:18 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.06.2020 23:01:13 EEST

ID користувача:  
91603

Назва документу: 1Нечипоренко\_КП-61

ID файлу: 1004021968 Кількість сторінок: 44 Кількість слів: 7944 Кількість символів: 60479 Розмір файлу: 97.80 KB

## 3.45% Схожість

Найбільша схожість: 0.54% з джерело <https://www.stud24.ru/advertising/ntegrovanij-pokaznik-efektivnost-reklamno-kam>

1.76% Схожість з Інтернет джерелами 23 ..... Page 46

2.29% Текстові збіги по Бібліотеці акаунту 83 ..... Page 46



**Дякую за увагу!**

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

\_\_\_\_\_ Іван ДИЧКА

«\_\_\_» \_\_\_\_\_ 2019 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ДЛЯ АНАЛІЗУ РЕКЛАМНОЇ КАМПАНІЇ**

**Програма та методика тестування**

ДП.045440-04-51

«ПОГОДЖЕНО»

Керівник проєкту:

\_\_\_\_\_ Леся ЛЮШЕНКО

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Владислав НЕЧИПОРЕНКО

## ЗМІСТ

1. Об'єкт випробувань.....	3
2. Мета тестування.....	3
3. Методи тестування.....	3
4. Засоби та порядок тестування.....	4

## **1. ОБ'ЄКТ ВИПРОБУВАНЬ**

Програмна забезпечення для аналізу рекламної кампанії являє собою мобільний додаток, створений на платформі iOS (мова програмування Swift) з використанням шаблону проектування MVC.

## **2. МЕТА ТЕСТУВАННЯ**

У процесі тестування має бути перевірено наступне:

- 1) відповідність елементів сторінок вимогам до мобільного застосунку;
- 2) наявність доступу до аналізу рекламної кампанії та продукції;
- 4) наявність обробки критичних ситуацій;
- 5) зручність роботи з мобільним застосунком;
- 6) відповідність дизайну вимогам Технічного завдання.

## **3. МЕТОДИ ТЕСТУВАННЯ**

Тестування виконується методом Black Box Testing. Перевіряється безпосередньо програмний продукт на відповідність функціональним вимогам з точки зору зовнішнього світу, без урахування знань про внутрішнє представлення системи. Тестування відбувається на рівні «системного тестування».

Використовуються наступні методи:

- 1) функціональне тестування, зокрема на рівні Critical path test (базове тестування);
- 2) тестування продуктивності програмного забезпечення, зокрема Stability testing (тестування стабільності) та Load testing (навантажувальне тестування);
- 3) тестування граничних значень;
- 4) тестування за допомогою припущення про помилку;
- 5) тестування інтерфейсу.

#### **4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ**

Тестування виконується засобами інструментарію Appium.

Працездатність програмної системи планування медичного обслуговування перевіряється шляхом:

- 1) динамічного ручного тестування – введенням граничних та недопустимих значень в поля, які можна редагувати;
- 2) динамічного ручного тестування на відповідність функціональним вимогам;
- 3) статичного тестування коду;
- 4) тестування мобільного додатка на пристроях різних моделей (з різними розмірами та апаратним забезпеченням);
- 5) тестування при максимальному навантаженні;
- 6) тестування стабільності роботи при різних умовах;
- 7) тестування зручності використання;
- 8) тестування інтерфейсу.



**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

«ЗАТВЕРДЖЕНО»

Науковий керівник кафедри

\_\_\_\_\_ Іван ДИЧКА

«\_\_\_» \_\_\_\_\_ 2020 р.

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**ДЛЯ АНАЛІЗУ РЕКЛАМНОЇ КАМПАНІЇ**

**Керівництво користувача**

ДП.045440-05-34

«ПОГОДЖЕНО»

Керівник проєкту:

\_\_\_\_\_ Леся ЛЮШЕНКО

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Владислав НЕЧИПОРЕНКО

## ЗМІСТ

1. Опис структури мобільного застосунку.....	3
2. Опис статичних сторінок.....	3
3. Процедура реєстрації чи авторизації користувача.....	11
4. Зміна даних в особистому кабінеті користувача.....	13
5. Процедура створення чи вибору файлу аналізу рекламної кампанії.....	15
6. Процедура детального перегляду графіків.....	18
7. Процедура аналізу рекламної продукції.....	20

## **1. Опис структури мобільного застосунку**

Програмне забезпечення для аналізу рекламної кампанії складається зі статичних сторінок та сторінок, вміст яких формується динамічно. Мобільний додаток виконаний українською мовою.

*Статичні сторінки мобільного застосунку:*

- меню мобільного застосунку;
- сторінка з аналізом рекламної кампанії;
- сторінка завершення процесу аналізу рекламної кампанії;
- журнал дій;
- інформація про застосунок.

*Динамічні сторінки мобільного застосунку:*

- особистий кабінет користувача;
- сторінка реєстрації чи авторизації користувача;
- сторінка створення чи завантаження файлу аналізу рекламної кампанії;
- сторінка з аналізом рекламної продукції;
- сторінка детального перегляду графіків;

Унизу мобільного застосунку знаходиться меню, за допомогою якого можна пересуватися між сторінками застосунку. Якщо сторінка не входить до цього меню, то на неї обов'язково знаходиться посилання на іншій сторінці й на ній самій знаходиться посилання на попередню сторінку.

У мобільному застосунку наявні 2 ролі користувача: «Admin» та «Звичайний користувач». «Admin» має доступ до журналу дій та може змінювати роль інших користувачів.

## **2. Опис статичних сторінок**

*Меню мобільного застосунку*

Меню мобільного застосунку (рис. 1) знаходиться внизу кожної основної сторінки мобільного застосунку (рис. 2), а саме: сторінки з

аналізом рекламної кампанії, сторінки з аналізом рекламної продукції, сторінки завершення процесу аналізу рекламної кампанії та особистого кабінету користувача. За допомогою нього можна пересуватися між цими сторінками. Синім кольором виділяється активна сторінка.

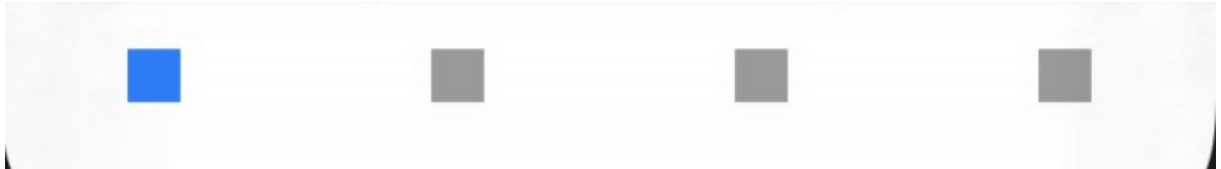


Рис. 1. Меню мобільного застосунку



Рис. 2. Вигляд меню на основній сторінці

### *Сторінка з аналізом рекламної кампанії*

На цій сторінці відображається процес аналізу рекламної кампанії (рис. 3-6). Ця сторінка є дуже великою і переміщатися по ній можна за допомогою прокрутки сторінки у вертикальному напрямку вгору чи вниз за допомогою пальця.

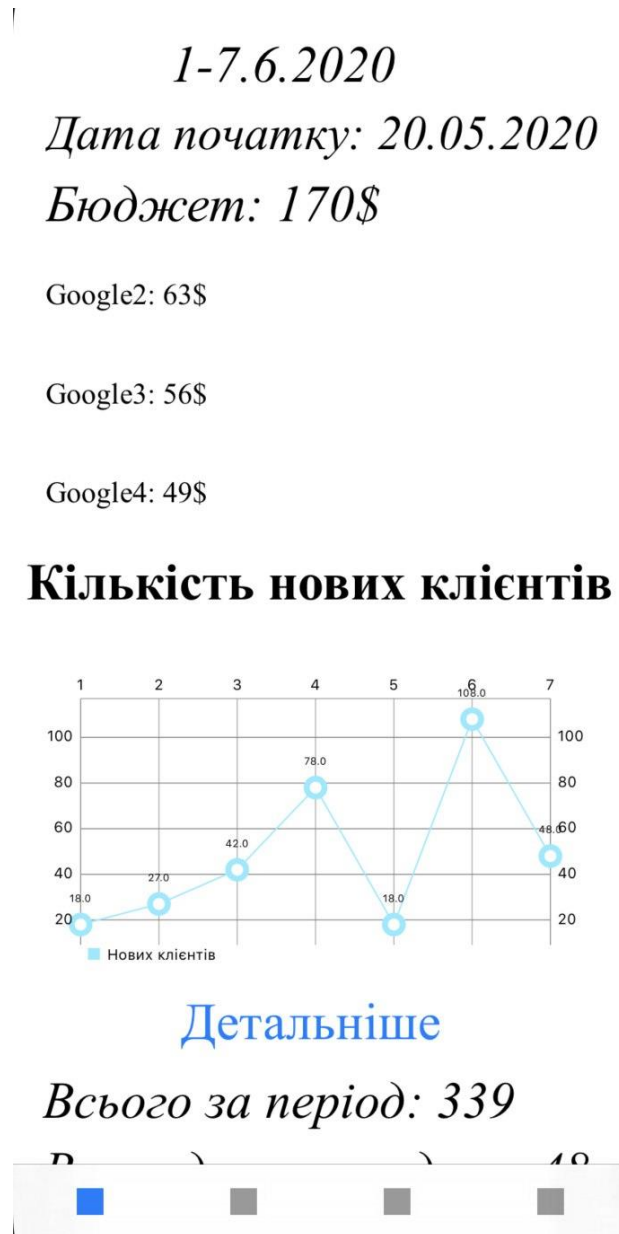
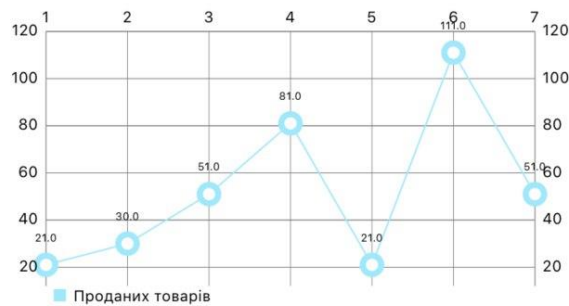


Рис. 3. Сторінка з аналізом рекламної кампанії

*Всього за період: 339*

*В середньому за день: 48*

### Кількість проданих товарів



### Детальніше

*Всього за період: 366*

*В середньому за день: 52*

### Прибуток

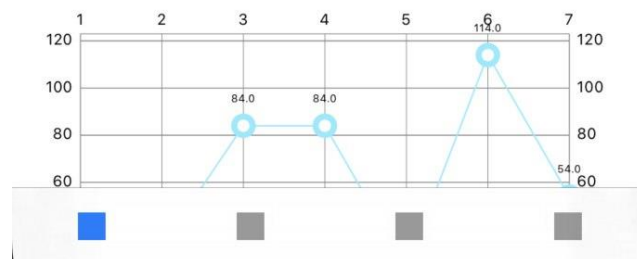
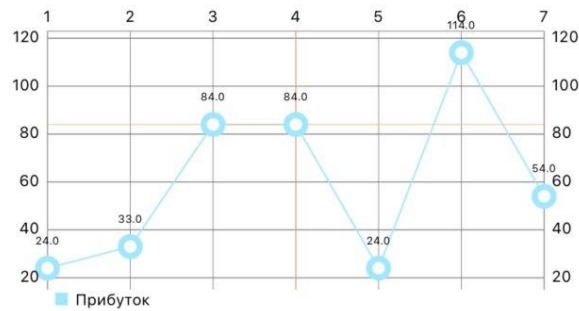


Рис. 4. Сторінка з аналізом рекламної кампанії

## Прибуток

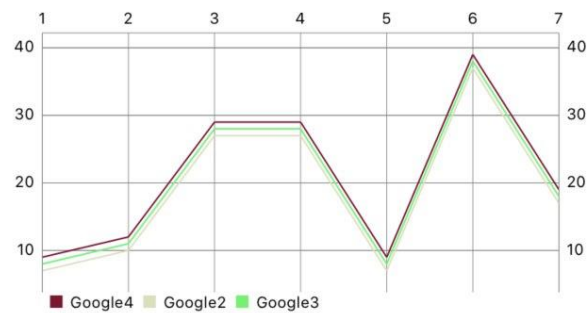


## Детальніше

*Всього за період: 417*

*В середньому за день: 59*

## Аналітика рекламних каналів



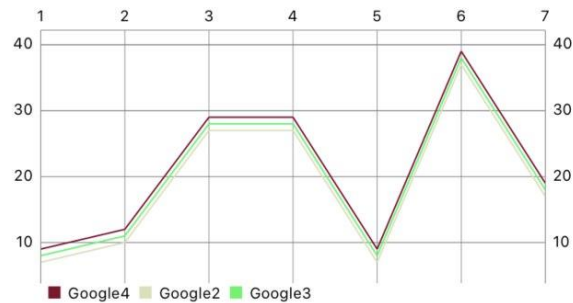
## Детальніше



Рис. 5. Сторінка з аналізом рекламної кампанії

*Всього за період: 417*  
*В середньому за день: 59*

### **Аналітика рекламних каналів**



[Детальніше](#)

*ROI: 26 %*  
*PPC: 0.97 \$*  
*CPO: 0.9 \$*  
*CPL: 1.23 \$*



Рис. 6. Сторінка з аналізом рекламної кампанії

*Сторінка завершення процесу аналізу рекламної кампанії*

Після того, як користувач натисне кнопку «Завершити», відобразиться інформація по завершенні рекламної кампанії (рис. 7).



**Найкращий рекламний канал**

**Google4**

**Найгірший рекламний канал**

**Google3**

*ROI: +23 %*

*PPC: +0.05 \$*

*CPO: +0.0 \$*

*CPL: +0.29 \$*

**Завершити**



Рис. 7. Сторінка завершення процесу аналізу рекламної кампанії

#### *Журнал дій*

На цій сторінці відображаються дії користувачів, а також можна змінити роль користувача (рис. 8).

## Журнал дій

2020/06/05 10:02:29 Vladislav Перейшов на сторінку аналізу рекламної кампанії

2020/06/05 10:02:30 Vladislav Перейшов на сторінку аналізу рекламної продукції

2020/06/05 10:02:31 Vladislav Перейшов на сторінку завершення аналізу рекламної продукції

2020/06/05 10:02:31 Vladislav Перейшов на сторінку аналізу рекламної продукції

2020/06/05 10:02:32 Vladislav Перейшов на сторінку аналізу рекламної кампанії

2020/06/05 10:02:33 Vladislav Перейшов на сторінку аналізу рекламної продукції

2020/06/05 10:02:33 Vladislav Перейшов на сторінку завершення аналізу рекламної продукції

2020/06/05 10:02:34 Vladislav Перейшов в особистий кабінет

[Попередня](#)

[Наступна](#)

[Admin](#) [Default](#) [Немає](#)

[Назад](#)

Рис. 8. Журнал дій

### *Сторінка з інформацією про застосунок*

На цій сторінці міститься необхідна інформація про розроблений застосунок – назва, версія, ім'я розробника та загальна інформація про розроблене програмне забезпечення (рис. 9).

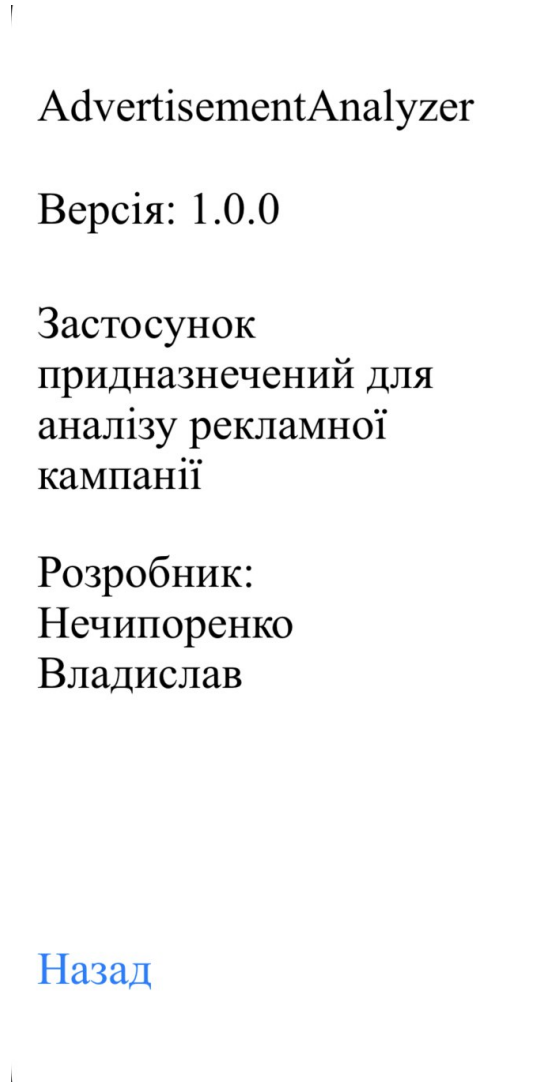


Рис. 9. Сторінка з інформацією про застосунок

### **3. Процедура реєстрації чи авторизації користувача**

Авторизація користувача відбувається на сторінці реєстрації чи авторизації (рис. 10). Користувач має ввести логін і пароль та натиснути кнопку «Далі». Якщо користувача з таким логіном не існує, він буде створений і відбудеться перехід на наступну сторінку. Якщо користувач з таким логіном існує і пароль вірний, також відбудеться перехід на наступну

сторінку. У випадку, коли користувач вводить невірну інформацію, виведеться відповідне повідомлення (рис. 11).

Реєстрація	Авторизація
Логін	<input type="text"/>
Пароль	<input type="password"/>
Ім'я	<input type="text"/>
Дата нар.	<input type="text"/>
Компанія	<input type="text"/>
<a href="#">Далі</a>	

Рис. 10. Сторінка реєстрації чи авторизації

Реєстрація

Авторизація

Логін

VladVlad

Пароль

VladVladddd

**Помилка!**

Такий логін вже існує і ви ввели невірний пароль

Ок

Далі

Рис. 11. Приклад повідомлення про помилку на сторінці реєстрації чи авторизації

#### 4. Зміна даних в особистому кабінеті користувача

У особистому кабінеті користувач має змогу переглядати та змінювати свою інформацію (рис. 12). Після зміни інформацію потрібно натиснути кнопку «Оновити» і в разі якщо вся інформація введена правильно дані оновляться. У випадку, коли користувач вводить невірну інформацію, виведеться відповідне повідомлення (рис. 13). Також, на цій сторінці знаходяться посилання на сторінку з інформацією про застосунок та журнал дій, доступний лише користувачам з роллю «Admin».

**Особистий кабінет**

Логін

Пароль

Ім'я

Дата нар.

Компанія

[Журнал дій](#)

[Інфо](#) [Оновити](#)

■ ■ ■ ■

Рис. 12. Особистий кабінет користувача

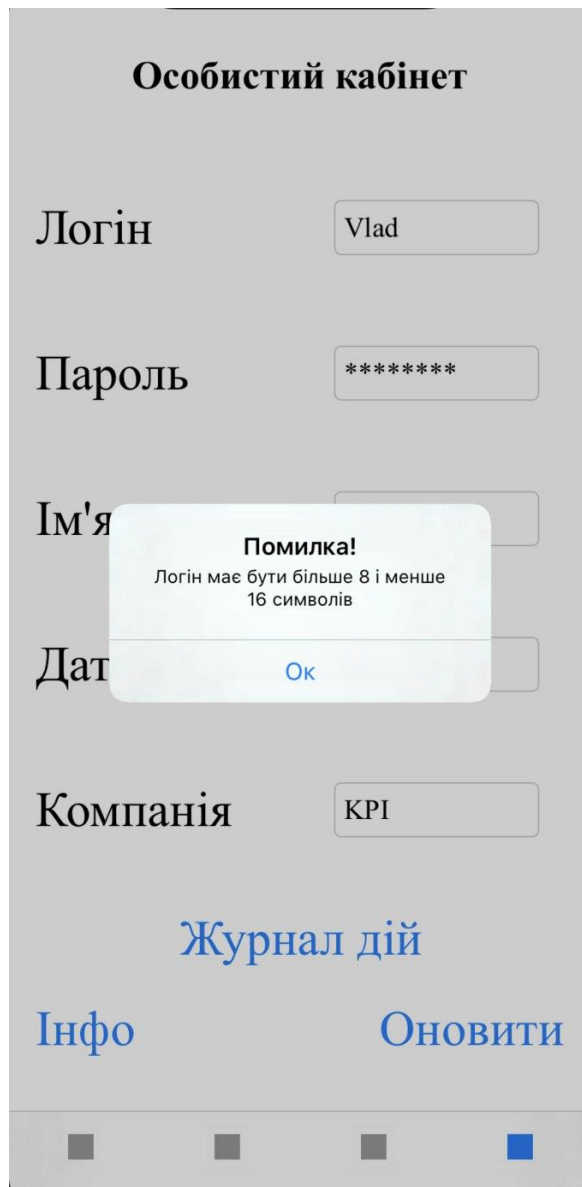


Рис. 13. Приклад повідомлення про помилку в особистому кабінеті

## 5. Процедура створення чи вибору файлу аналізу рекламної кампанії

На сторінці створення чи завантаження файлу аналізу рекламної кампанії необхідно ввести всю інформацію та натиснути кнопку «Далі» (рис. 14). Якщо інформація введена правильно відбудеться перехід на сторінку з аналізом рекламної кампанії. У випадку, коли користувач вводить невірну інформацію, виведеться відповідне повідомлення (рис. 15).

Створити

Завантажити

Назва

Бюджет

Джерело

Шлях

Назви полів:

Дата

Назва джерела

Нових клієнтів

Проданих товарів

Прибуток

Витрати

Назва продукту

Назва БД

Назва таблиці

Далі

Рис. 14. Сторінка створення чи завантаження файлу аналізу рекламної кампанії



Створити

Завантажити

Назва

name

Бюджет

dada

Джерело

MongoDB

Шлях

name

Назви полів:

Дата

date

Назва

Новий

Продукт

Прибуток

profit

Витрати

waste

Назва продукту

name

Назва БД

name

Назва таблиці

name

Далі

Помилка!

У поле бюджет необхідно вводити число

Ok

Рис. 15. Приклад повідомлення про помилку на сторінці створення чи завантаження файлу аналізу рекламної кампанії

## 6. Процедура детального перегляду графіків

На сторінці детального перегляду графіків необхідно вибрати усю необхідну інформацію з випадаючих списків та натиснути кнопку «Далі» (рис. 16). Після чого зміни відобразяться на графіку (рис. 17). Якщо ж натиснути кнопку «Оновити», значення показників оновляться (рис.18). У випадку, коли користувач вводить невірну інформацію, на графіку буде відображатися відповідне повідомлення (рис. 19).

Рік	<input type="text"/>
Місяць	<input type="text"/>
Тиждень	<input type="text"/>
Показник	<input type="text"/>
Джерело	<input type="text"/>

*ROI:*

*PPC:*

*CPO:*

*CPL:*

[Назад](#) [Оновити](#) [Далі](#)

Рис. 16. Сторінка детального перегляду графіків

[Назад](#)

## Прибуток Google2

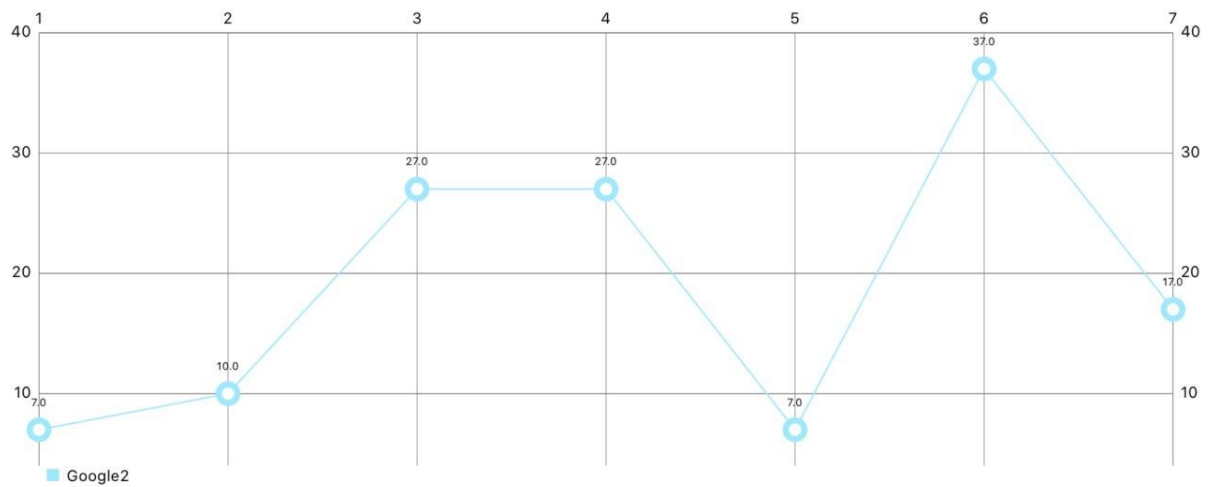


Рис. 17. Графік

Рік	2020	▼
Місяць	5	▼
Тиждень	4	▼
Показник	Прибуток	▼
Джерело	Усього	▼

*ROI: 3 %*

*PPC: 1.06 \$*

*CPO: 0.98 \$*

*CPL: 1.1 \$*

[Назад](#) [Оновити](#) [Далі](#)

Рис. 18. Оновлені дані

[Назад](#)

**Прибуток**

Немає даних



Рис. 19. Графік у разі відсутності даних

## **7. Процедура аналізу рекламної продукції**

На сторінці аналізу рекламної продукції необхідно ввести всю необхідну інформацію та натиснути кнопку «Оновити» (рис. 20). Після чого зміни відобразяться на графіку (рис. 21).

8-14.6.2020

## Запитів

Немає даних

Джерело

Назва

Детальніше

Оновити

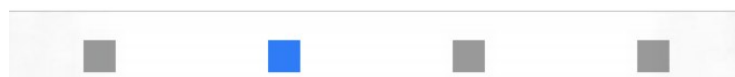
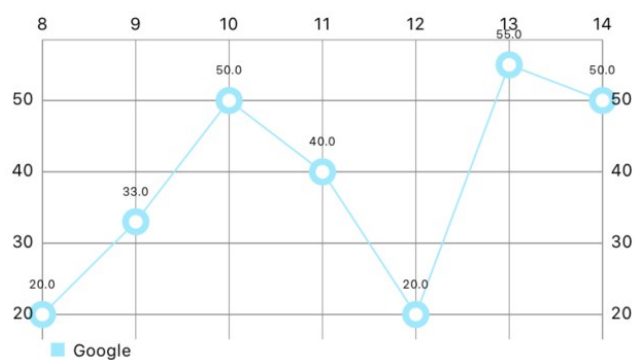


Рис. 20. Сторінка аналізу рекламної продукції

8-14.6.2020

## Запитів



Джерело

Google ▼

Назва

MyProduct

Детальніше

Оновити



Рис. 21. Графік аналізу рекламної продукції